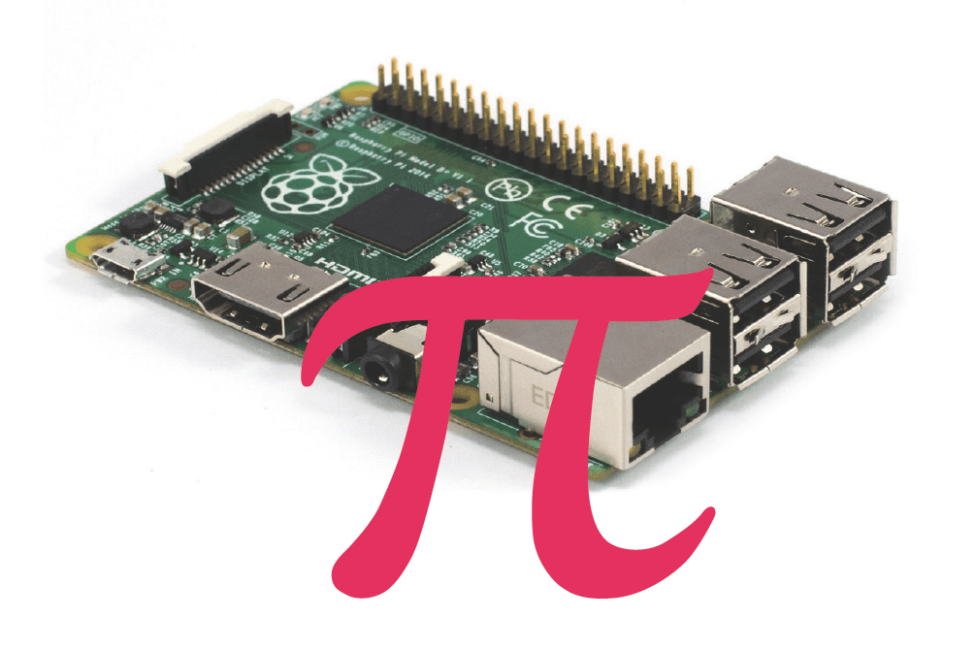
柯博文老师首本系统论述树莓派开发理论、技巧及案例实战的百科全书!

简体中文版、繁体中文版、英文版三种版本全球同步发售!





The Practical Developing Guide for Raspberry Pi
Learning 100 Excellent Projects Step by Step

树莓派（Raspberry Pi）实战指南

手把手教你掌握100个精彩案例

柯博文◎编著
Powen Ko



清華大学出版社

清华开发者书库

树莓派(Raspberry Pi)实战指南
——手把手教你掌握100个精彩案例

柯博文 编著

清华大学出版社
北 京

内 容 简 介

本书系统介绍了开源硬件设备树莓派(Raspberry Pi)的架构、编程、开发与项目实践。全书共23章,内容包括搭建网络、网站服务器系统、家庭云存储系统与私有云、机顶盒、运行Android系统、LCD点阵式液晶气象预报机、物联网系统、照相机与摄像机设计、多媒体播放机、增强现实影像识别系统、条码扫描机、掌上游戏机、遥控器开关、遥控汽车、无线路由器、机器人、超级计算机群网等。

为了提高学习效果,本书为所有案例设计提供了完整的教学录像及源代码文件。

本书适合作为高校嵌入式课程的教材或教学参考书,也可供广大电子爱好者、开源硬件发烧友阅读。

图书在版编目(CIP)数据

树莓派(Raspberry Pi)实战指南:手把手教你掌握100个精彩案例/柯博文编著.--北京:清华大学出版社,2015
(清华开发者书库)
ISBN 978-7-302-40118-6

Ⅰ.①树… Ⅱ.①柯… Ⅲ.①Linux操作系统—指南 Ⅳ.①TP316.89-62

中国版本图书馆CIP数据核字(2015)第089504号

责任编辑: 盛东亮
封面设计: 李召霞
责任校对: 李建庄
责任印制: 沈 露

出版发行: 清华大学出版社
网 址: http://www.tup.com.cn, http://www.wqbook.com
地 址: 北京清华大学学研大厦A座 **邮 编:** 100084
社 总 机: 010-62770175 **邮 购:** 010-62786544
投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn
课件下载: http://www.tup.com.cn,010-62795954
印 刷 者: 北京鑫丰华彩印有限公司
装 订 者: 北京市密云县京文制本装订厂
经 销: 全国新华书店
开 本: 186mm×240mm **印 张:** 25.25 **字 数:** 582千字
版 次: 2015年6月第1版 **印 次:** 2015年6月第1次印刷
印 数: 1~2500
定 价: 79.00元

产品编号:064108-01

前言
PREFACE

当前,Raspberry Pi 已经是非常稳定的革命性产品。据国外主流媒体报导,自 2012 年 2 月推出到 2015 年 1 月止,Raspberry Pi 已在全球销售了 350 万台,这个惊人的数字毫无疑问地确立了它在计算机发展史上的地位。Raspberry Pi 可以运用于计算机初学者的教育、低收入者的个人计算机、无人机控制系统、海洋探索设备……甚至是 Amazon 自动送货飞机的核心设备。这一切都是因为它体积小、功耗低、开源,而且吸引了无数开发者投入其中并乐于分享。

通过多台 Raspberry Pi 的协助应用,可以预见未来云计算和分布式计算的强劲发展;而其价格低、体积小的优势,更为物联网的发展提供了实用的解决方案。每一天,都有来自世界各地源源不断的创意涌现,相信 Raspberry Pi 的未来还有更多发挥想象力探索的空间。2015 年 2 月,Raspberry Pi 的新版本 Raspberry Pi2 Model B(900MHz Quard Core)问世,运行速度更为出色,运算效果更为优异。

笔者因为居住地的关系,很早就有机会把玩与探索 Raspberry Pi 在各领域的应用,并多次在我国台湾新竹工研院的公开课程、企业内部训练课程和高校授课过程中与大家分享 Raspberry Pi 的魅力。现在,借助此书,与大家进一步系统分享 Raspberry Pi 与 Linux 系统在物联网、无人载具、大数据、机器人、影像识别、智能电视等方面的应用。

感谢清华大学出版社的鼎力协助,让这本书顺利出版上市。更感谢广大读者的认可,让我更有动力分享创新科技。目前,笔者在全球各地都定期讲授课程,读完此书后想进一步深造的读者,可以访问笔者的个人网站或者报名参加相关课程。若有培训需求,欢迎您来信至 powenkoads@gmail. com。

由于笔者居住在美国硅谷近 20 年,简体中文表达功底不足,书中如有表达不清楚或笔误之处,也欢迎您来信或到网站上留下您的宝贵意见,我会尽可能一一回复。

最后,祝大家在 Raspberry Pi 的学习和使用中,一帆风顺,马到成功。

柯博文
于美国硅谷 San Jose

学习说明

STUDY SHOWS

本书视频课堂地址

书中提及的完整教学影片可到51CTO学院观看学习，网址：

http://edu.51cto.com/lecturer/user_id-8340419

工程文件下载地址

清华大学出版社网站本书页面，网址：

www.tup.tsinghua.edu.cn

本书作者联络方式

柯博文的网站：http://www.powenko.com

柯博文的脸书：http://www.facebook.com/powenko1

柯博文的微博：http://t.sina.com.cn/powenko

本书读者对象包括

电子爱好者、嵌入式开发者、开源硬件玩家、树莓派发烧友……

本书提供以下案例

网络搭建、网站服务器系统、家庭云存储系统与私有云、机顶盒、运行Android系统、点阵式液晶气象预报机、物联网系统、照相机与摄像机设计、多媒体播放机、增强现实影像识别系统、条码扫描机、掌上游戏机、遥控器开关、遥控汽车、无线路由器、机器人、超级计算机群网……

目 录
CONTENTS

第1章 树莓派

1.1 树莓派介绍

2012年2月29日诞生的树莓派,它的出现绝对有资格在计算机发展史中写下浓重的一笔。在多年前,很难相信一台个人计算机居然会缩小到一个名片大小的尺寸,并且能够实现所有的功能,而树莓派就让此事成真。和现在的平板电脑不一样的是,树莓派底层是一个完整的Linux操作系统,而不像Android的底层还是一个精简过的Linux嵌入式版本。所以说可以在个人计算机上执行的Linux程序,几乎都可以在树莓派上执行。它是当今表现最让人惊艳的计算机。

大部分计算设备(包括手机,平板电脑和游戏机)都是一体成形的,很难在上面开发与设计相关的应用软件,并且也很难去修改硬件,而树莓派却是完全相反的。看到设计精致的绿色电路板那一刻起,它就邀请你一同进入它的万能计算机世界,并且一切都是公开的,所有开发者都乐于把信息与大家分享。树莓派是一个开放源代码的硬件平台,该平台包括一块具备简单I/O功能的电路板及Broadcom BCM2835 system on a chip (SoC)的ARM芯片,并且拥有一大堆的Linux软件。树莓派可以用来开发交互产品,例如它可以读入大量的开关和感测器信号,并且可以控制电灯、电机和其他各式各样的物理设备。树莓派可以开发出与PC一样的周边设备,也可以执行在Linux PC上的软件进行网络通信。树莓派的硬件电路板可以自行焊接组装,也可以购买已经组装好的电路板,程序开发环境的软件可以从网上免费下载与使用。重点是硬件电路板的价格非常便宜,今天的市价约180元人民币。它配备了需要的工具,也可以开发新的软件和程序,并且可以连接到它自己的电子硬件,用它来设计周边硬件,截至2014年2月,已经有250万个开发板售出。

1.2 树莓派模块B

1.2.1 树莓派模块B

2012年2月,树莓派基金会发布一款新产品,名称为Raspberry Pi Model B(树莓派模

块 B)。当前在美国的定价是 29.99 美金(2015 年 2 月),外形如图 1-1 所示。

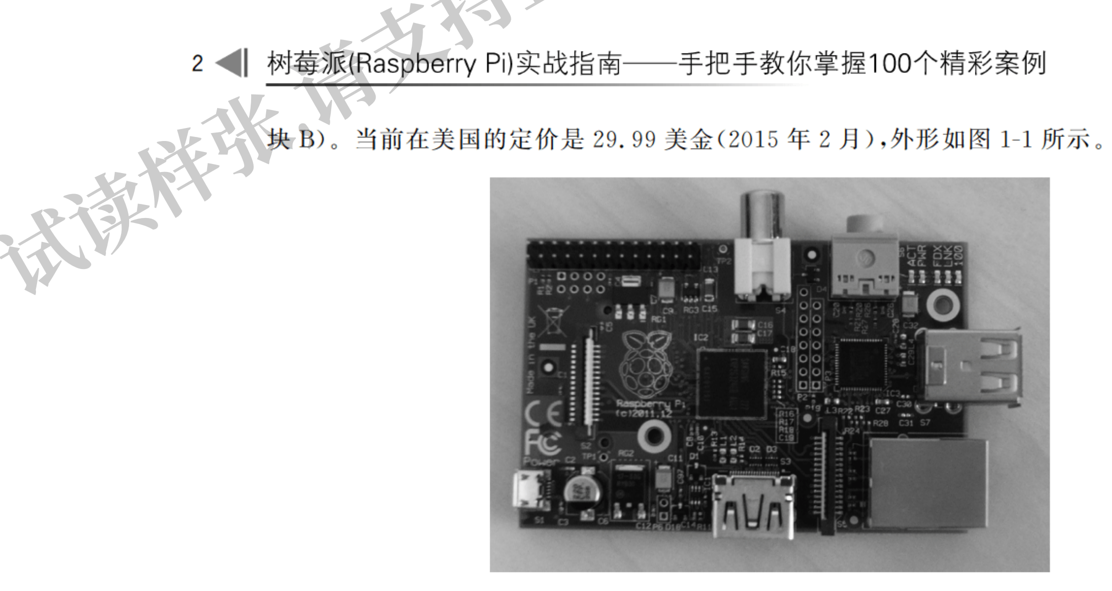


图 1-1　树莓派模块 B 的外形

树莓派模块 B 功能非常强大,它的几大功能如图 1-2 所示。

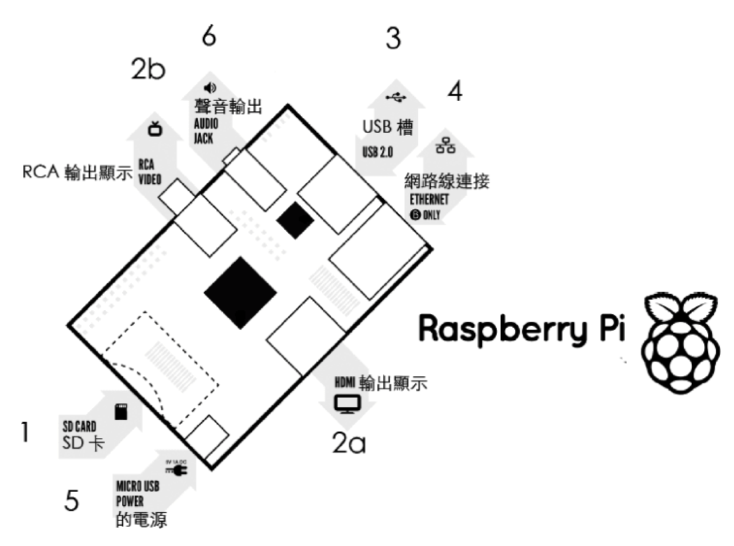


图 1-2　树莓派模块 B 的功能

(1) SD 卡:树莓派模块 B 会把 SD 卡当成内置的硬盘使用,一般 SD 卡最少需要 4GB,推荐使用比较稳定的牌子,这样可以确保读入数据正常。

(2) 2a:HDMI 输出显示。2b:RCA 输出显示。

(3) USB 的输入连接,如鼠标与键盘。

(4) RJ45 网络线连接。

(5) 通过 Micro USB 的电源:电源最少需要 700mA/5V。

（6）Audio Jack 声音输出。

可以通过表 1-1 了解树莓派的硬件规范，当前市面上也有很多加强版本和变形版本，这本书的内容是以官方的树莓派 rev 2 Model B 为主。

表 1-1　树莓派的硬件

项目	说　明
开发者	树莓派基金会
式样	单一板子的计算机
上市时间	2012 年 2 月 29 日
建议价格	25 美金（Model A）和 35 美金（Model B）
操作系统	Linux（Raspbian、Debian GNU/Linux、Fedora and Arch Linux ARM）、RISC OS、FreeBSD、NetBSD、Plan 9
额定功率	2.5W（Model A）和 3.5W（Model B）
CPU	ARM1176JZF-S（armv6k）700MHz
存储	SD 或者 SDHC 卡
内存	256MB（Model A） 512MB（Model B rev 2） 256MB（Model B rev 1）
GPU	Broadcom VideoCore IV
官方网站	www.raspberrypi.org

这个树莓派的硬件板子右上角有几个 LED 灯，如图 1-3 所示。当出现以下情况时，LED 指示灯会亮。

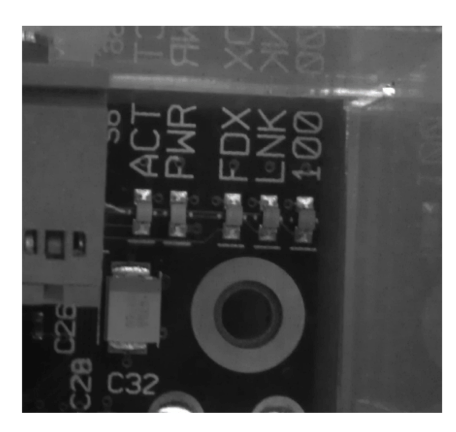


图 1-3　硬件 LED 灯

（1）ACT：SD 卡处于活动状态。

（2）PWR：打开 3.3V 电源。

（3）FDX：网络 adapter 处于全双工（Full Duplex Connection）状态。

(4) LNK:网络连接成功。

(5) 100:网络速度是100Mbps。

1.2.2 树莓派模块 B+

2014年7月,树莓派基金会发布一款改良版的产品,名称为"Raspberry Pi Model B+"(树莓派模块B+),外形如图1-4所示。

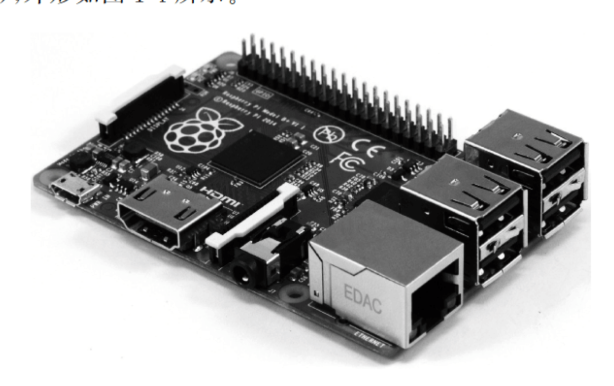


图1-4 树莓派模块B+的外形

这个版本的特点如下:

(1) 更低功耗;

(2) 改善的电源管理系统;

(3) 将输入/输出的GPIO的引脚从原本的26个提高到40个;

(4) USB Ports接头从原本的两个提升到4个;

(5) 把原本的RCA屏幕接头改成4针的连接头;

(6) SD卡升级到MicroSD。

1.3 树莓派计算机模块

树莓派已经成为一个新势力,并有成千上万专业的产品正在使用。但还是有些地方不是那么符合工业产品的专业市场标准。为了解决这些问题,树莓派基金会在2014年4月,发布了一个新的模块Raspberry Pi Compute Module(树莓派计算机模块),如图1-5所示,现在已经可以在市面上买到了。

树莓派计算机模块的尺寸与一个内存模块SODIMM的尺寸接近,提供了完整的树莓派的功能,但它的尺寸只有原来树莓派的四分之一。为何要重新设计呢?主要原因是SODIMM是一种成本低廉并且常用的尺寸标准,通过模块的设计可以把IO的界面独立出来。针对不同的情况,可以使用不同的模块。

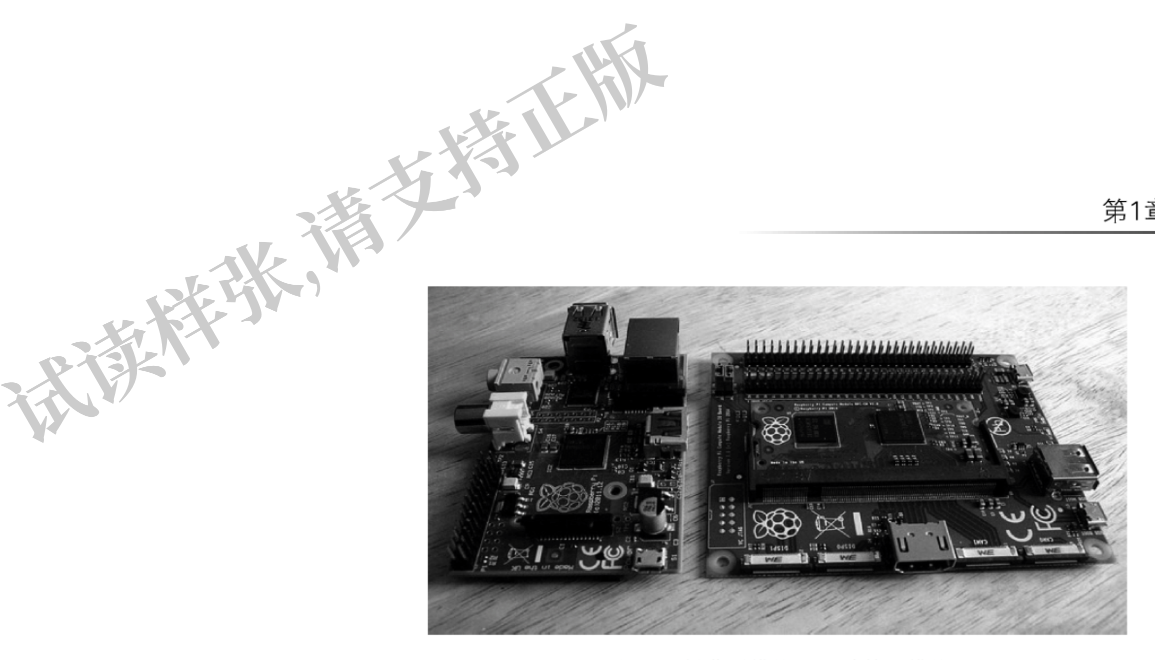


图 1-5　树莓派模块 B 和计算机模块

1. 树莓派 Computer Module

树莓派 Computer Module 是计算机模块主板，如图 1-6 所示，其功能如下：

(1) SODIMM 内存插槽是长为 6.5cm，宽为 3cm 的树莓派板子。

(2) 使用的是 BCM2835 芯片，内存为 512MB 的 RAM(Random Access Memory 即随机存取存储器)。

(3) 在板子上还有 4GB eMMC Flash Memory 快闪存储器，用来存储开机时的操作系统。

(4) 提供的 200 个引脚，用来连接实现 BCM2835 芯片的所有功能。

(5) 使用的是家庭应用的 FCC 的 B 类认证。

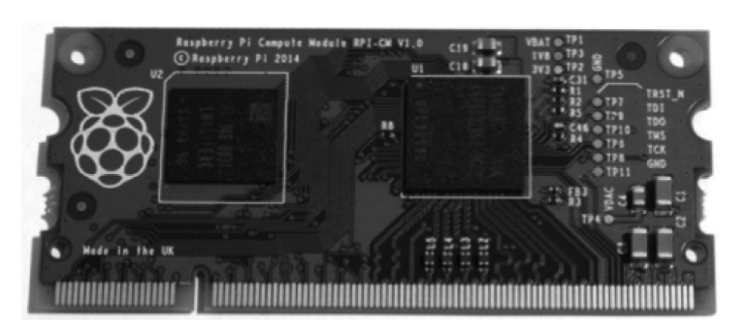


图 1-6　SODIMM 造型的树莓派 Computer Module

2. 树莓派 IO Board

树莓派 IO Board 是输入/输出实验板，如图 1-7 所示，其功能如下：

(1) 两个各 60 个的引脚，可以程序化控制的引脚是 GPIO1 至 GPIO45。

(2) 两个 Micro USB type B。

(3) 1 个 USB type A。

(4) 两个 CSI 接口给摄像机。

(5) 两个 DSI 接口给显示面板。

(6) 1 个 HDMI 接口。

(7) 1 个 Micro USB 的电源接头。

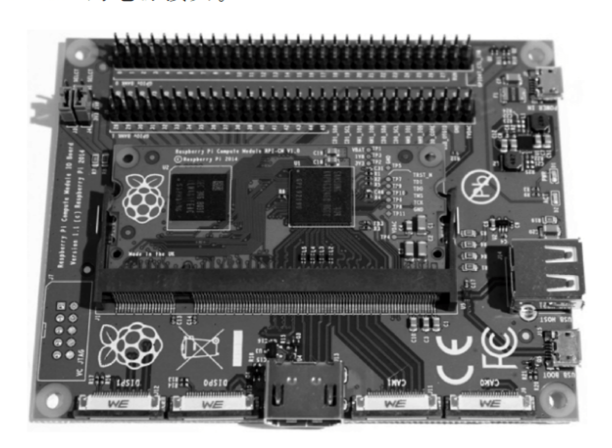


图 1-7 树莓派 IO Board

将树莓派 Computer Module 和 IO Board 板子连接起来,成为一整套,其外观如图 1-8 所示。



图 1-8 树莓派 Computer Module 和 IO Board 连接起来外观

1.4 树莓派 2

新的树莓派 2 在 2015 年 2 月正式发布,其外观如图 1-9 所示。它让很多 DIY 玩家眼前一亮,树莓派 2 在保持原定的 $35 美金售价不变的前提下,性能大幅提升,甚至具备运行定

制版的 Windows10 系统的潜力。

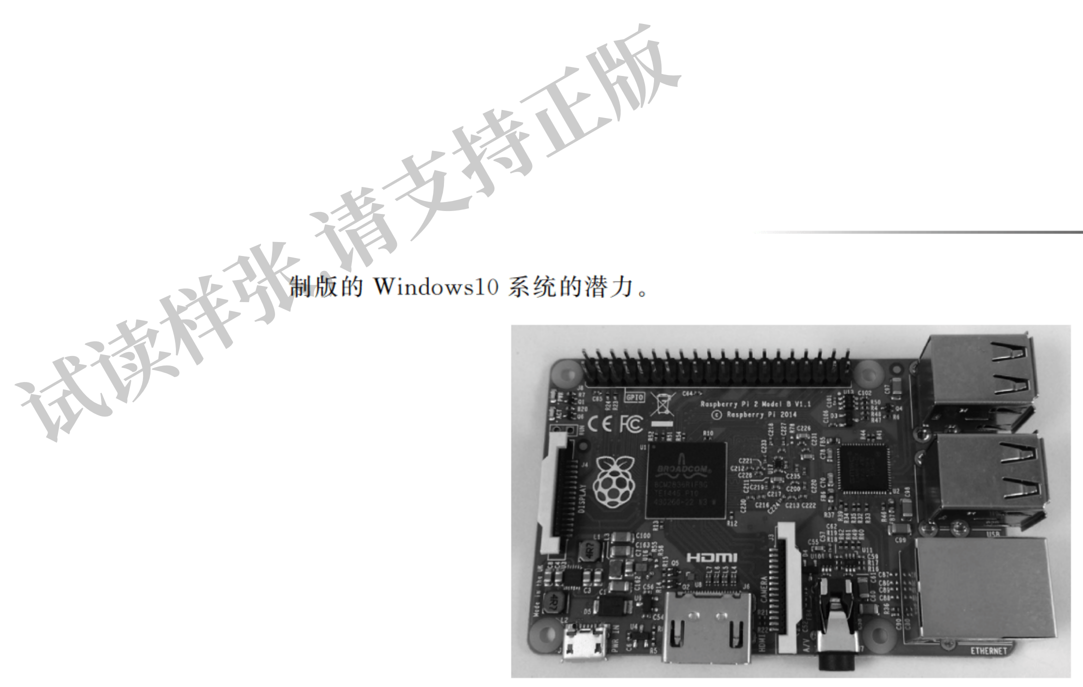


图 1-9 树莓派 2 外观

树莓派 2 的硬件配置如下:

(1) 处理器为博通 BCM2836 ARMv7 的四核处理器,运行频率为 900MHz。

(2) 内存为 1GB RAM。

(3) GPIO 的引脚延续前一版的样式。

(4) 支持完全兼容 HAT。

(5) 具有 40PIN 的 GPIO 引脚。

(6) 具有 10/100 以太网端口。

本版的速度提高了 6 倍,比上一版具备更好地反映和处理速度,书中介绍的 OS 都能顺利在树莓派 2 上运行。

第 2 章　树莓派的操作系统

树莓派当前有多个操作系统,并且在陆续增加中。本章将介绍有哪些操作系统可以使用,下面是当前可以执行在树莓派上的操作系统。

1. Raspbian

这是当前使用最广的树莓派的操作系统,如图 2-1 所示。本书大部分的案例是用这个版本来介绍的。

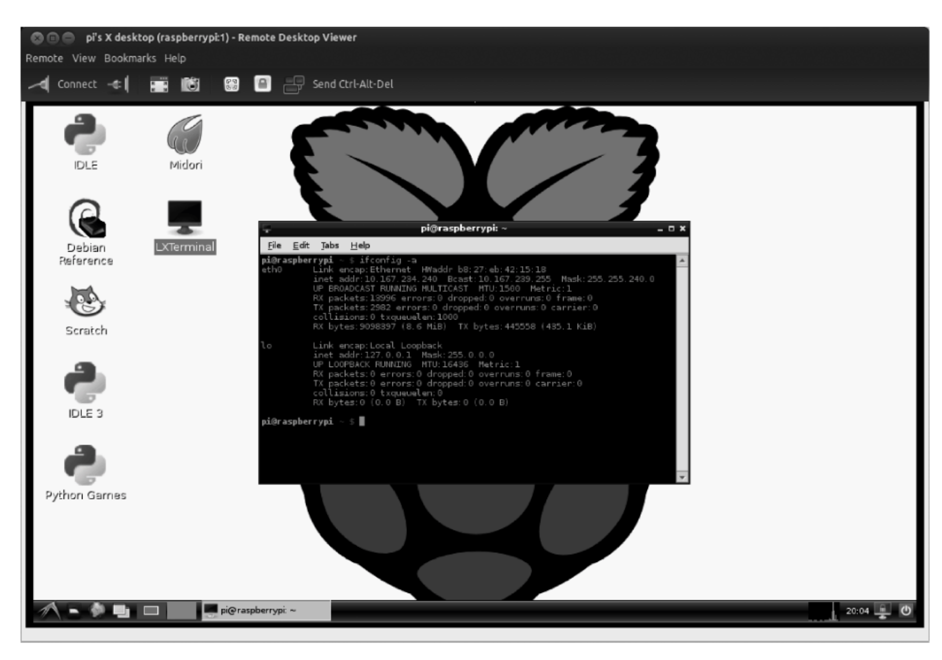


图 2-1　Raspbian 在树莓派上运行

2. Arch Linux 树莓派版本

Arch Linux 也是树莓派的操作系统,如图 2-2 所示,对 Linux 操作系统很熟悉的人大都使用此版本。



图 2-2　Arch Linux 树莓派版本在树莓派上运行

3. Pidora 树莓派版本

Pidora 树莓派版本是 Linux 操作系统修改后的版本,如图 2-3 所示,是拥有另外一种风格的树莓派操作系统。



图 2-3　Pidora 树莓派版本在树莓派上运行

4. Risc 树莓派版本

Risc 树莓派版本是功能丰富的树莓派的 Linux 操作系统，如图 2-4 所示。

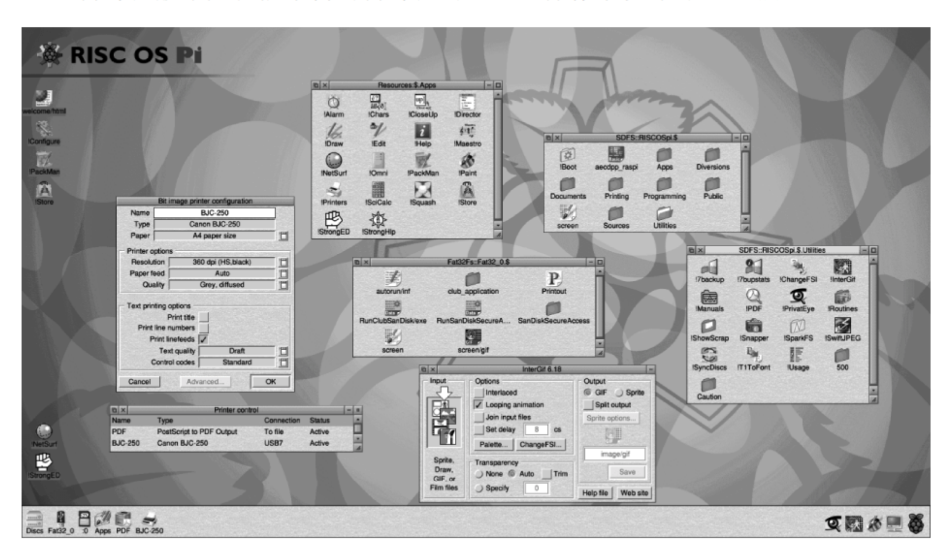


图 2-4　Risc 树莓派版本在树莓派上运行

5. Raspbmc

如果想设置树莓派成为电视的操作系统，Raspbmc 是一个很棒的选择，如图 2-5 所示，它支持很多多媒体音视频格式和网站，整体搭建完毕后，会让树莓派像智能电视一样。

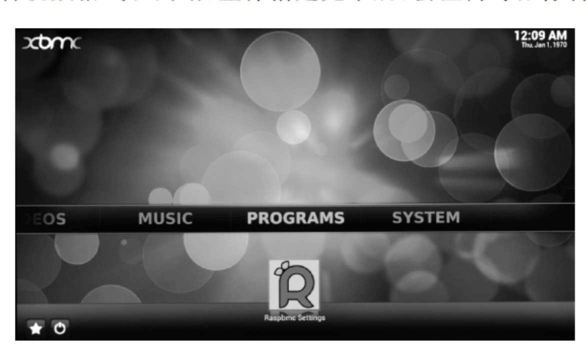


图 2-5　Raspbmc 在树莓派上运行

6. Android 树莓派版本

当前已经公布可以顺利地在树莓派上执行的 Android 版本是 2.3，如图 2-6 所示，但是执行速度和性能还需要再进一步改进，另外 4.0 版本也正在移植修改中。

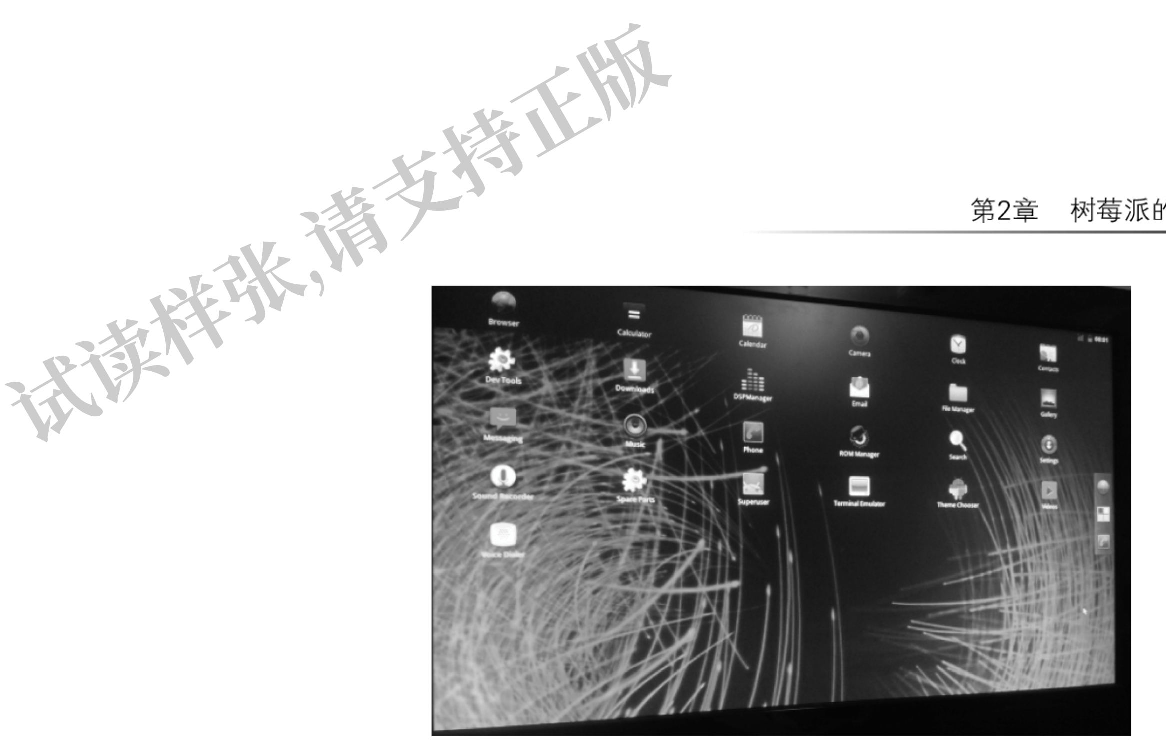


图 2-6 Android 树莓派版本在树莓派上运行

7. OpenElec

Open Embedded Linux Entertainment Center(OpenElec)是一个小的 XBMC 媒体操作系统,如图 2-7 所示。

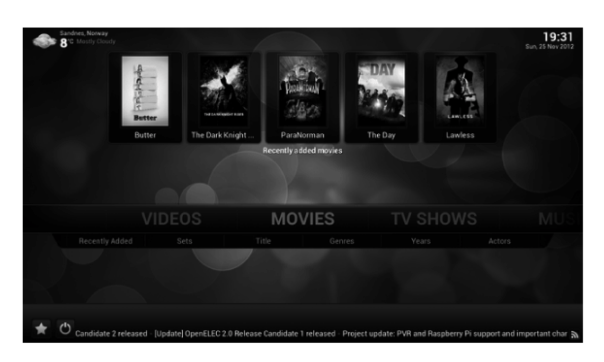


图 2-7 OpenElec 在树莓派上运行

2.1 搭建树莓派 NOOBS 图形操作系统

案例 1

1. 介绍

一步一步地介绍如何把树莓派顺利开机和引导。

本书主要介绍多个树莓派的应用和案例的高级应用,如果觉得学习上有困难,推荐先阅读柯老师的著作,碁峰资讯出版的《树莓派最佳入门与实战应用——深入树莓派的全方面的

指南》。

2. 目的

在树莓派上安装与搭建 NOOBS 图形操作系统。

3. 设备

需要有以下硬件:

(1) 树莓派板子;

(2) 树莓派兼容的 SD 卡;

(3) USB 的电源线;

(4) RJ45 网络线;

(5) SD 读卡机。

4. 接线

首先把网线和 USB 电源线接到树莓派硬件上,然后把 SD 卡读卡机插到 Windows 计算机上。

5. 步骤

STEP1:下载树莓派的操作系统

树莓派有几个操作系统可供选择,本章介绍最受大家喜爱的 Raspbian,在计算机上通过浏览器连接到 http://www.raspberrypi.org/downloads,如图 2-8 所示,下载最新 NOOBS 的 zip 版本。

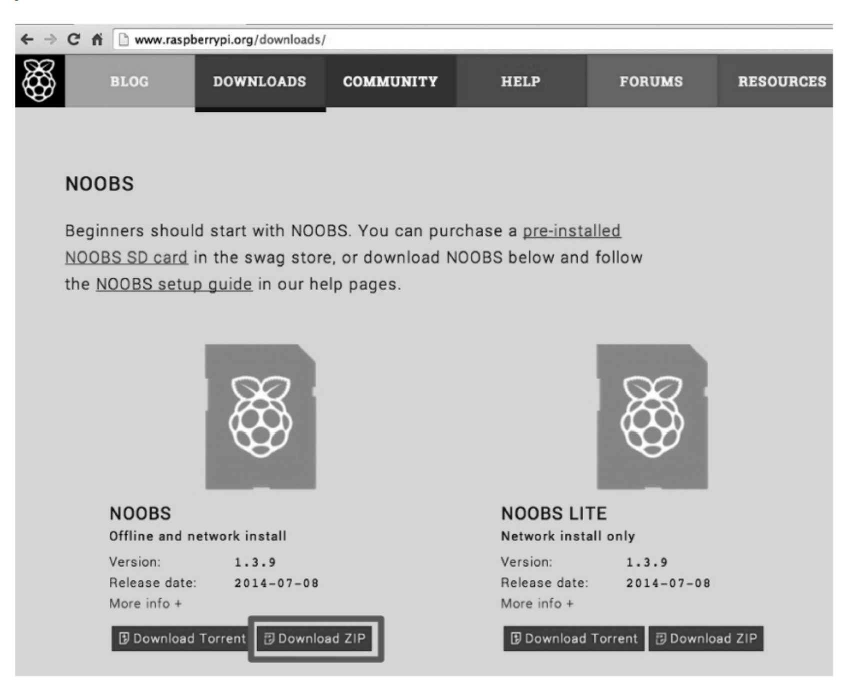


图 2-8 下载最新版本

STEP2：解压缩

下载成功后，因为是 zip 的压缩文件，如果在 Windows 的操作系统上，可以单击鼠标右键，选择“解开文件 Extract all”，会看到.img 的文件，这个.img 文件就是镜像文件。

STEP3：下载 SD Format 4.0 格式化工具

首先把 4GB 以上容量的 SD 卡备份数据，做格式化的准备，在 Windows 的操作系统上，请到 https://www.sdcard.org/downloads/formatter_4/eula_windows/下载最新版本的 SD Format 4.0 格式化工具，如图 2-9 所示。

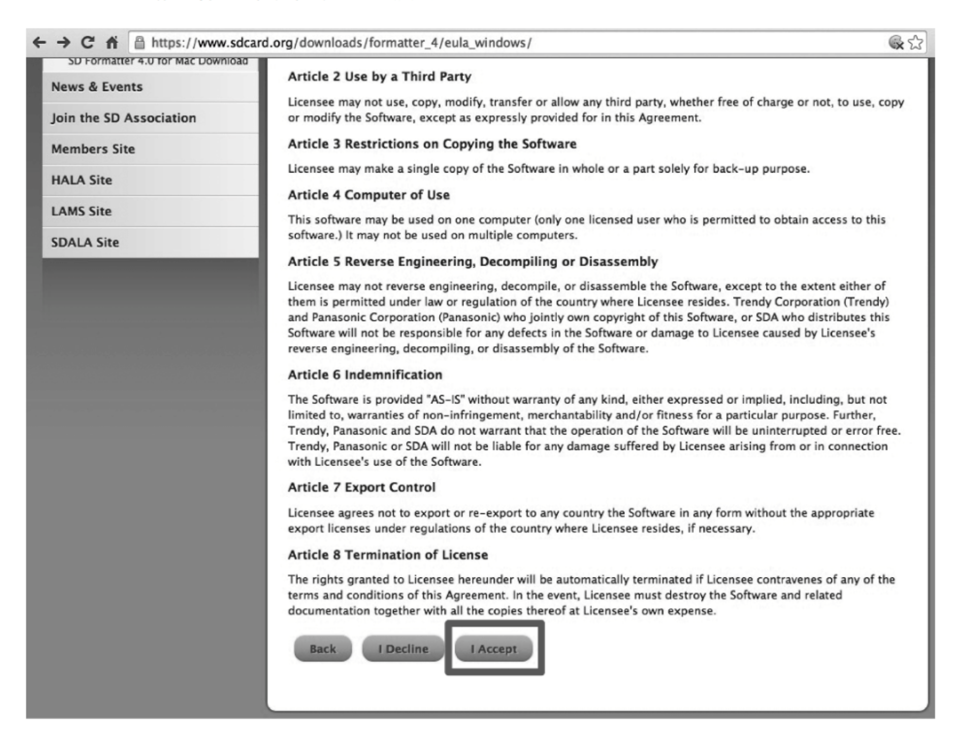


图 2-9 下载 SD Format 4.0 格式化工具

STEP4：格式化 SD 卡

下载后，请执行 SD Format 4.0 格式化软件，依照图 2-10 进行处理。

(1) 单击 Option 按钮，弹出 Option Setting 框，将 FORMAT SIZE ADJUSTMENT 选项设置为“ON”。

(2) 插入 SD 卡。

(3) 单击 Format 按钮，格式化 SD 卡。

STEP5：复制 NOOBS 文件到 SD 卡的根目录

打开文件夹，把刚刚下载和解压缩的 NOOBS 复制到 SD 卡的根目录，如图 2-11 所示。

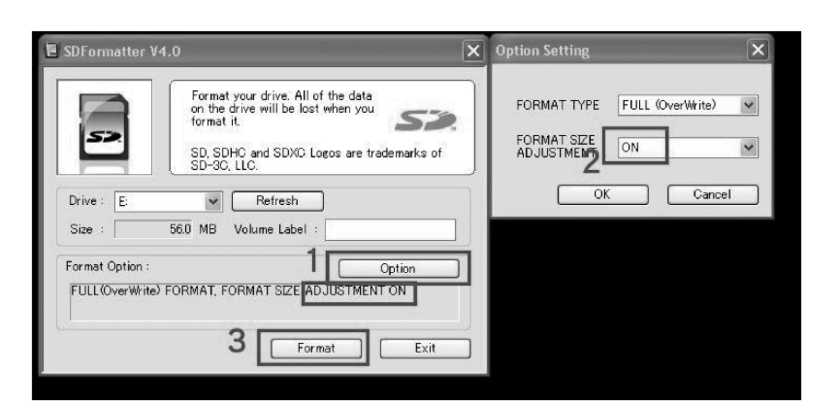


图 2-10　格式化 SD 卡

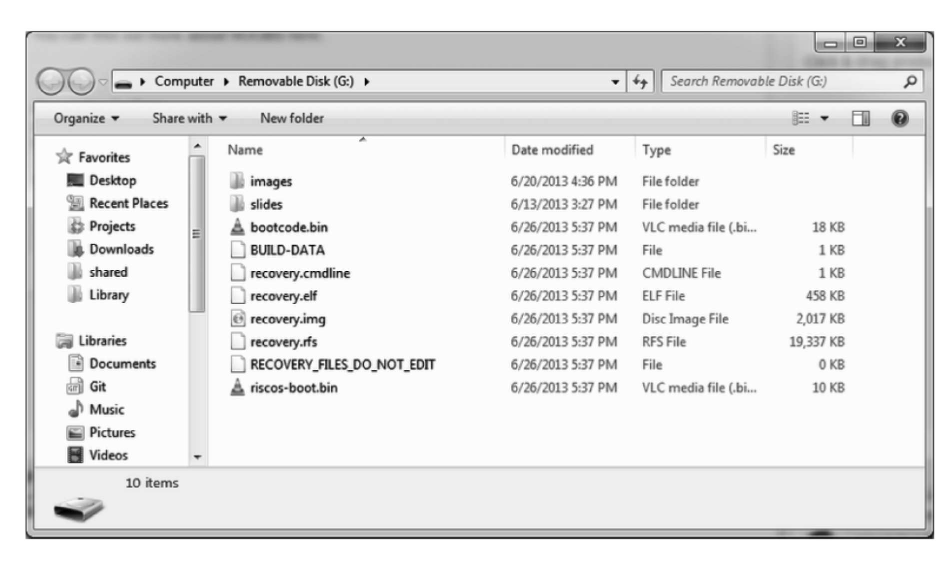


图 2-11　复制 NOOBS 文件到 SD 卡的根目录

STEP6：开机

在树莓派接上电源前，第一次开机使用前，请依照以下顺序把相关的东西连接起来。

(1) 使用 HDMI 线连接显示器和树莓派，如果显示器不支持 HDMI 接口，可以购买转换头，把 HDMI 接口转换成 VGA 接口。

(2) 连接 USB 鼠标。

(3) 连接 USB 键盘。

(4) 把已准备好的 SD 卡，插入树莓派上(不要接反了)。

(5) 连接 RJ45 接口网线。

(6) 最后把 mini USB 接到树莓派上,另外一头接到计算机上,或者接到所谓的 USB 供电器(例如:智能手机的充电器)上。

6. 结果

如果在屏幕上看到如图 2-12 所示,恭喜已顺利开机成功!

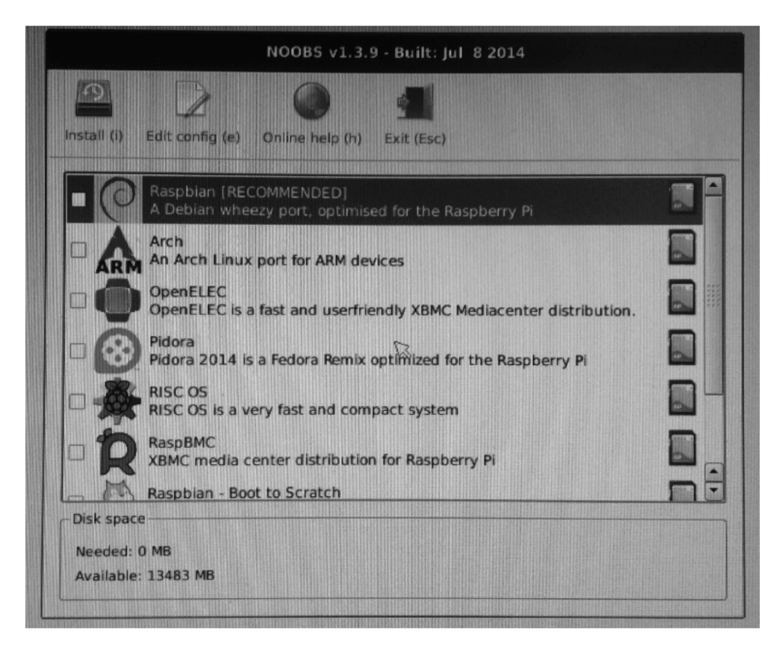


图 2-12　在屏幕上看到系统出现,代表开机成功

2.2　搭建树莓派 Raspbian 图形操作系统

案例 2

1. 目的

严格来说 NOOBS 还不算是操作系统,它算是协助安装其他操作系统的工具,所以下面试试实际安装一个完整的操作系统 Raspbian。

2. 步骤

STEP1:设置

继续 2.1 节,请在 NOOBS 做以下设置,如图 2-13 所示。

(1) 单击选中 Raspbian。

(2) 单击 Language 下拉框,选中 English(UK)选项。

(3) 单击 Keyboard 下拉框,选中 gb 选项,最后单击 Install 按钮。

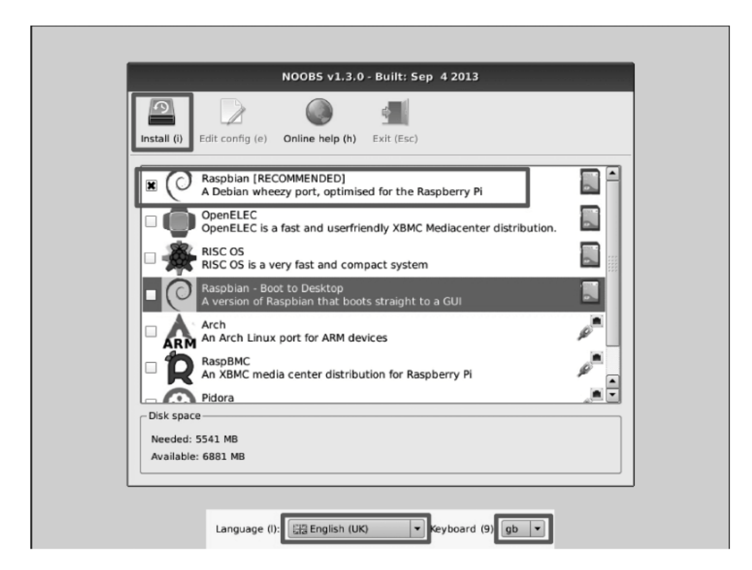


图 2-13　设置和安装 Raspbian

STEP2：确定

然后 NOOBS 系统会弹出警告框，如图 2-14 所示，确认是否安装新的操作系统，会覆盖 SD 卡中的所有数据，单击 Yes 按钮，以确认安装操作。

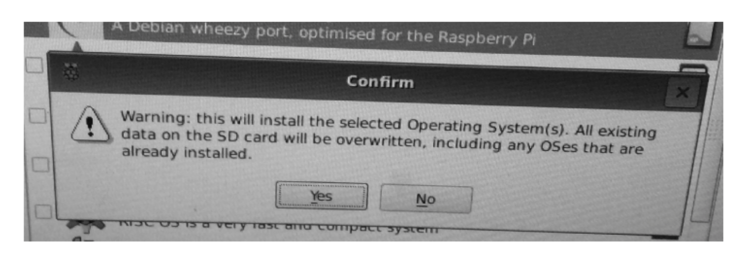


图 2-14　警告信息

STEP3：安装

NOOBS 会自动下载安装新的系统，如图 2-15 所示。

3. 结果

在屏幕上看到如图 2-16 所示的信息，代表顺利安装成功 Raspbian。系统默认的账号密码是

账号：pi

密码：raspberry

4. 教学影片

完整的教学视频可以看 raspberryPi_NOOBS_setup。

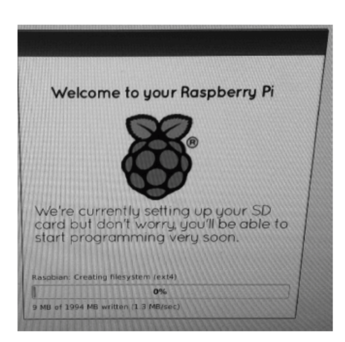


图 2-15　下载和安装 Raspbian



图 2-16　Raspbian 安装成功

2.3　Raspbian 的设置与引导

案例 3

1. 步骤

STEP1：进入设置

当第一次打开 Raspbian，正常应出现如图 2-17 所示的设置界面，如果没有出现，在命令行中输入以下指令，就可以进入 Raspbian 系统的设置界面。

```
$ sudo raspi-config
```

STEP2：扩充 SD 卡

第一次使用推荐选择“1. Expend Filesystem”来扩充 SD 卡的空间。

STEP3：键盘设置

虽然已经在 NOOBS 中设置过键盘了，但还是再确认和学习一次如何修改键盘设置，请选择 Internationalisation Options→Change Keyboard Layout，然后在键盘设置指定为 Generic 105-key，如图 2-18 所示。

STEP4：键盘设置选项

请依照如图 2-19 所示进行处理。

(1) 键盘 layout 设置，请指定为 Other。

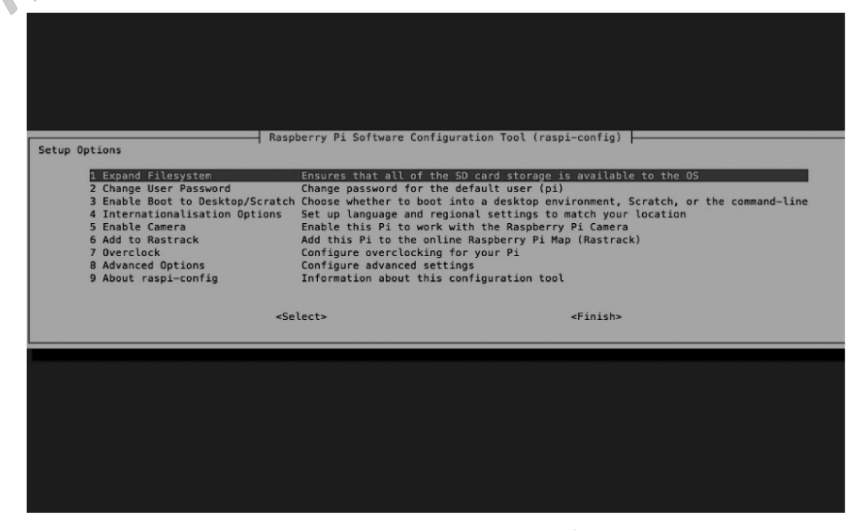


图 2-17 sudo raspi-config 进入环境设置

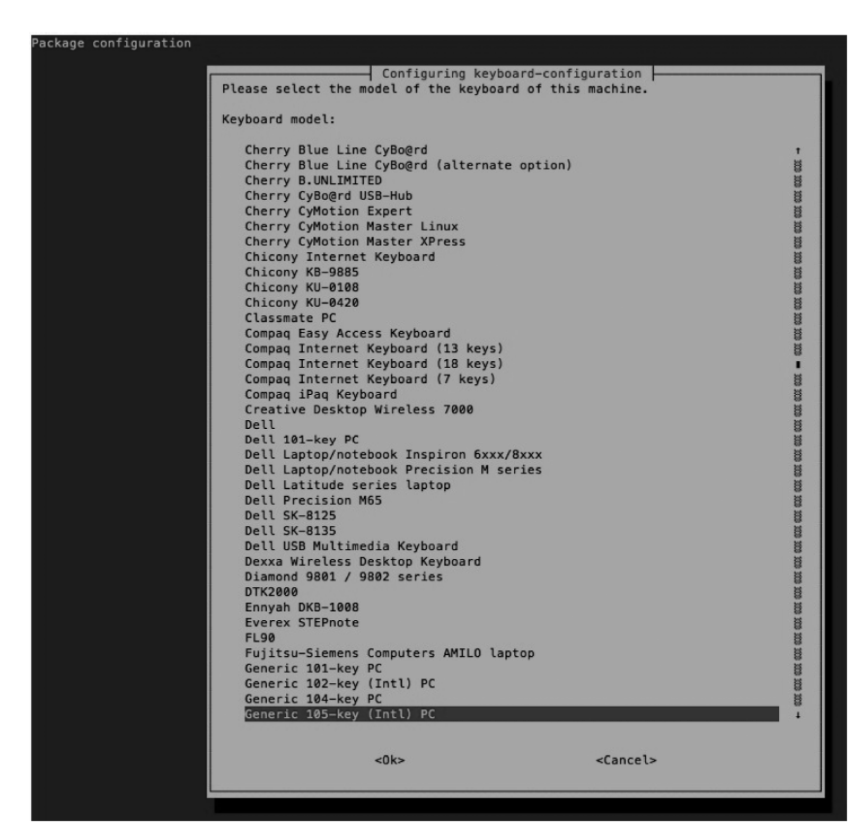


图 2-18 键盘设置指定为 Generic 105-key



图 2-19　键盘 layout 设置

(2) 设置键盘的国家或地区时,请指定为 English (US)。

(3) 设置键盘的 Layout 时,请指定为 English (US)。

(4) 设置键盘的 Key to function as AltGr 时,请指定为 Both Alt keys。

(5) 设置键盘的 Compose Key 时,请指定为 No compose key。

(6) 设置是否可以用 Control+Alt+Backspace 来离开 x Server? 请选中 Yes。

STEP5:键盘设置完成

回到设置的主选项,选择 final 离开后,键盘就会正常了。

2. 教学影片

完整的教学视频可以看 raspberryPi_Raspbian_init。

第 3 章 网络搭建

3.1 设置固定 IP 网络

案例 4

1. 介绍

因为树莓派是使用动态的 IP 网络地址,这样每次都得通过 ifconfig 命令确认网络位置。本章会介绍如何设定网络的固定 IP,以方便下次使用时,不再询问系统现在的网络地址,并且也强烈推荐设置固定 IP 地址。后面的网络案例,例如搭建网站服务器、个人云等,并且对有些软件,使用固定 IP 地址都会比较方便。

2. 步骤

STEP1:用文本编辑器设置 etc/network/interface

因为下面的设置需要输入固定的网络地址、gateway、netmask、networkc 和 broadcast。如果不了解目前的网络设置,可以通过如图 3-1 所示的命令,查询这个地址并记录下来。

```
$ ifconfig
```

```
pi@raspberrypi /etc/network $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:76:58:3e
          inet addr:192.168.0.111  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1213 errors:0 dropped:0 overruns:0 frame:0
          TX packets:339 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:85876 (83.8 KiB)  TX bytes:46265 (45.1 KiB)
```

图 3-1 查询 IP 地址、broadcast(Bcast)地址、netmask(Mask)地址

通过如图 3-2 所示的命令,获取 gateway(就是 default via 后的信息,即是图 3-2 中的 192.168.0.1)。

```
$ ip route show
```

```
pi@raspberrypi /etc/network $ ip route show
default via 192.168.0.1 dev eth0
192.168.0.0/24 dev eth0  proto_kernel  scope link  src 192.168.0.111
```

图 3-2　获取 gateway

STEP2：用文本编辑器设置 etc/network/interfaces

在文本模式下输入以下的命令，就会进入 nano 文本编辑模式，编辑/etc/network/interfaces 文件。

```
$ sudo nano /etc/network/interfaces
```

以下是原本的网络设置，系统自动获取 IP 地址。

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

STEP3：修改固定 IP 地址

请按以下的数据输入和修改 eth0（网卡）的设置，就是第 4 行至第 8 行，如果在家中，address(固定 IP 地址)就可以设置 192.168.1.3 至 192.168.1.253 之间的地址。其他部分，请按照 STEP1 所得的地址填入。如果该设备在公司或学校，请和网管人员沟通，他们会给你合适的地址。同样的道理设置 wlan0（无线网卡 WiFi）。

```
1.  auto lo
2.  iface lo inet loopback
3.
4.  iface eth0 inet static
5.  address 192.168.0.111
6.  gateway 192.168.0.1
7.  netmask 255.255.255.0
8.  broadcast 192.168.0.255
9.
10. allow-hotplug wlan0
11. iface wlan0 inet manual
12. wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
13. iface default inet dhcp
```

然后，按下 Control+o+Enter 组合键存储文件，并且按下 Control+x 组合键离开 nano 文本编辑器，如图 3-3 所示。

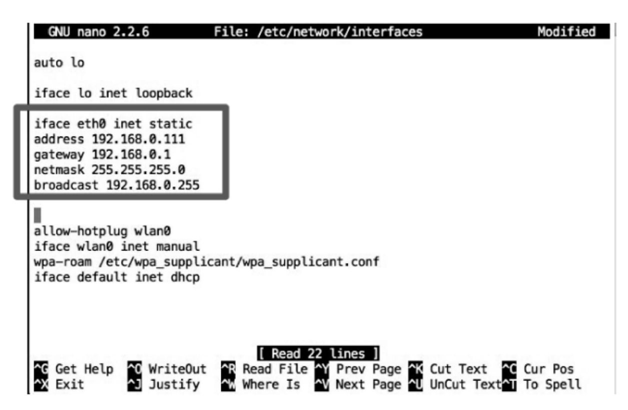


图 3-3　设置固定 IP 的/etc/network/interfaces 文件

STEP4：修改固定 IP 地址

请输入以下命令重新开机。

```
$ sudo reboot
```

开机后，输入以下命令确认 IP 地址。

```
$ ifconfig
```

3.2　设置无线 WiFi

3.2.1　挑选 WiFi 设备

案例 5

1. 目的

要让树莓派可以直接连接 WiFi，但是树莓派对使用 USB 的 WiFi 设备还是很严格的，树莓派对 WiFi 网卡也挑剔，所以推荐不要乱买市面上的网卡，否则会找不到驱动程序。

2. 挑选 WiFi 硬件

请使用官方推荐的 WiFi 网卡 Edimax EW-7811Un 150 Mbps Wireless 11n Nano Size USB Adapter with EZmax Setup Wizard。如果还想知道支持哪些 WiFi 设备，可以查看 http://elinux.org/RPi_USB_Wi-Fi_Adapters，简要列举如下：

```
•  7DayShop
       W - 3S01BLK, W - 3S01BLKTWIN:
       AWUS036NEH:
       AWUS036NH:
```

AWUS036NH:
AWUS036H
AWUS036H
AWUS036NHA:
AWUS036NHR:
AirLink101

- Alfa
  AWLL5088:
  AWLL5099:
  AWLL6075:
- Asus
  USB－N10 USB ID 0b05
  USB－N13 USB ID 0b05:17ab
  WL－167G v1 USB ID 0b05:1706
  WL－167G v3 USB ID 0b05:1791
- Edimax
  Edimax EW－7811Un 150 Mbps Wireless 11n Nano Size USB Adapter
- Linksys
  Linksys (Cisco) WUSB100 ver.2 1737:0078、Linksys (Cisco) WUSB600N
  Linksys WUSB54GC、Linksys WUSB54G ver.4,

确认网卡驱动程序,安装后,重新开机,使用 lsusb 命令确认网卡驱动程序安装成功,如图 3-4 所示。

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 7392:7811 Edimax Technology Co., Ltd EW-7811Un 802.11n Wireless Adapter [Realtek RTL8188CUS]
```

图 3-4　通过 lsusb 命令查看驱动程序是否安装成功

```
$ lsusb
```

3.2.2　设置 WiFi 密码方法一：通过 wpa_gui

案例 6

1. 目的

这里介绍两种方法,都可以设置 WiFi 的密码和指定连接的 WiFi router(路由器),首先介绍最容易的方法,通过图形化窗口 wpa_gui 来设置。

2. 步骤

STEP1：窗口环境

进入窗口环境。

```
$ startx
```

STEP2：打开 wpa_gui

接下来在“开始”菜单中选择 internet→wpa_gui，用来打开网络设置的窗口 wpa_gui，如图 3-5 所示。



图 3-5　选择 internet→wpa_gui

STEP3：寻找 WiFi router

单击 Scan 按钮，扫描出有多少 WiFi router 可以供网络连接，如图 3-6 所示。

STEP4：选择要连接的 WiFi router

接下来会出现 Scan results 窗口，选中要连接的 WiFi router，双击名称就可以进入下一个步骤，如图 3-7 所示。



图 3-6　单击 Scan 按钮

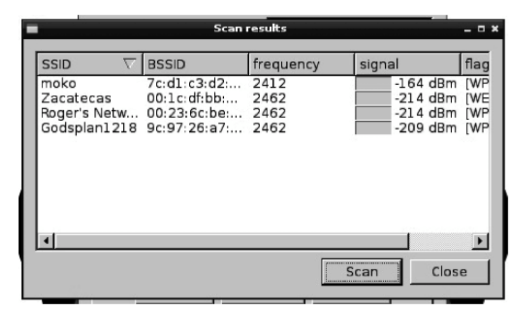


图 3-7　选择要连接的 WiFi router

STEP5：设置连接密码

选择正确的认证方式(Authentication)，设置密码(PSK)，完成后单击 Add 按钮，关闭此窗口，如图 3-8 所示。



图 3-8　设置连接密码

STEP6：连接和测试

回到 wpa_gui 窗口，选择正确的网络（network），也就是刚刚所设置要连接的对象 WiFi router，单击 Connect 按钮，如果一切顺利就会看到 IP address 项显示网络地址，如图 3-9 所示。注意完整的连接过程，需要 10 秒钟左右，请耐心等待。



图 3-9　单击 Connect 按钮

STEP7：完成

如果在图 3-9 中，已经看到 IP address 的地址，密码设置即已完成。

3. 教学影片

完整的教学视频可以看 raspberryPi_Setup_WiFi。

3.2.3 设置 WiFi 密码方法二：通过文件编辑

案例 7

1. 目的

另一个方法，通过网络配置文件/etc/network/interfaces 调整连接的 WiFi router 网络。

2. 步骤

STEP1：修改网络配置文件

编辑网络配置文件。

```
$ sudo nano /etc/network/interfaces
```

修改第 6～10 行的 wlan0 无线网卡的部分，eth0 是网卡的部分。

```
1.  auto lo
2.
3.  iface lo inet loopback
4.  iface eth0 inet dhcp
5.
6.  allow-hotplug wlan0
7.  auto wlan0
8.  iface wlan0 inet dhcp
9.  wpa-ssid "ssid"
10. wpa-psk "password"
```

以上设置的意思是

设置 WiFi AP 的 SSID(Service Set Identifier 服务集标识符)，例如 WiFi router 名为 powenko：

```
wpa-ssid "powenko"
```

设置 WiFi 密码，例如 WiFi 密码为 powenko.com：

```
wpa-psk "powenko.com"
```

所以请按照你的 WiFi 网络,设置 WiFi 的 SSID 和密码。

eth0 指的是网卡的设置,dhcp 是动态 IP 地址,如果想把它设置为固定 IP 地址,请按照本书的“3.1 设置固定 IP 网络”来修改以下的设置:

```
iface eth0 inet static
address 192.168.x.x
gateway 192.168.x.x
netmask 255.255.x.x
broadcast 192.168.x.x
```

同样的方法,如果网卡的部分也要用固定 IP,请把

```
iface wlan0 inet dhcp
```

改成

```
iface wlan0 inet static
address 192.168.x.x
gateway 192.168.x.x
netmask 255.255.x.x
broadcast 192.168.x.x
```

这里的 x.x 网址请按照实际使用的网络情况,自行调整为合适的网络地址。

完成后,在编辑环境下按下 CTRL＋o 组合键存储,CTRL＋x 组合键离开 nano 编辑器。

STEP2:重新开机

重启机器就可以使用了,或者可以输入以下命令重启网络设置。推荐重启机器操作,以下的命令执行后,可能会出现不生效的情况。

```
$ sudo /etc/init.d/networking restart
```

STEP3:确认

使用以下命令确认是否已经安装成功,并且查询无线网卡取的 IP 地址,如图 3-10 所示。

```
$ sudo ifconfig
```

```
pi@raspberrypi ~ $ sudo ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:2f:d2:ca
          inet addr:192.168.0.114  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2990 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1002 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:206398 (201.5 KiB)  TX bytes:131076 (128.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 80:1f:02:d3:8a:40
          inet addr:192.168.0.115  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3513 errors:0 dropped:4 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1024305 (1000.2 KiB)  TX bytes:1838 (1.7 KiB)
```

图 3-10 wlan0 有网络地址就代表设置成功

3.3 在树莓派上搭建 VNC Server

案例 8

1. 介绍

如何设置 VNC Server 远程控制树莓派？VNC 和 SSH 远程连接有何不同？VNC 方便进入 startx 窗口模式，SSH 只能看到纯文本模式。搭建好 VNC Server，如果多台树莓派同时运行，可以不用每一台都有键盘、鼠标和显示器等硬件设备。顺便说一下，树莓派在 Raspbian 的搭建与设置部分，SSH 已经搭建好了(纯文本模式的远程连接控制功能)。

2. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 网络连接设备。

需要有以下软件：

VNC Server

3. 接线

无

4. 步骤

STEP1：更新安装软件

通过以下命令，更新安装软件。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

这用来更新安装程序的设置。

STEP2：安装 tightvncserver

通过以下命令安装 tightvncserver。

```
$ sudo apt - get install tightvncserver
```

STEP3：引导 tightvncserver

通过以下命令引导 tightvncserver，第一次引导时会询问远程连接的密码，设置一组 8 个字节以上的密码就可以了，如图 3-11 所示。

```
$ tightvncserver
```

引导 session。

```
$ vncserver :1 - geometry 1024x728-depth 24
```

```
pi@raspberrypi ~ $ tightvncserver
You will require a password to access your desktops.

Password:
Warning: password truncated to the length of 8.
Verify:
```

图 3-11 引导 tightvncserver

3.4 在计算机上使用 VNC Client 远程控制树莓派

案例 9

1. 介绍

在 PC、Mac、Linux 上有很多 VNC Viewer 软件，如果你之前都没有使用过，推荐使用 REAL VNC 这一款免费的软件，如图 3-12 所示。下载地址是 http://www.realvnc.com/download/viewer/。

下载之后，填上实际的 IP 地址和在 Server 上的编号。然后单击 Connect 按钮就可以了，如图 3-13 所示。

第一次使用时，VNC Viewer 会弹出提示警告消息，直接单击 Continue 连接。如果看到桌面程序的界面，就代表顺利成功。

2. 教学影片

完整的教学视频可以看 raspberrypi_VNCServer。

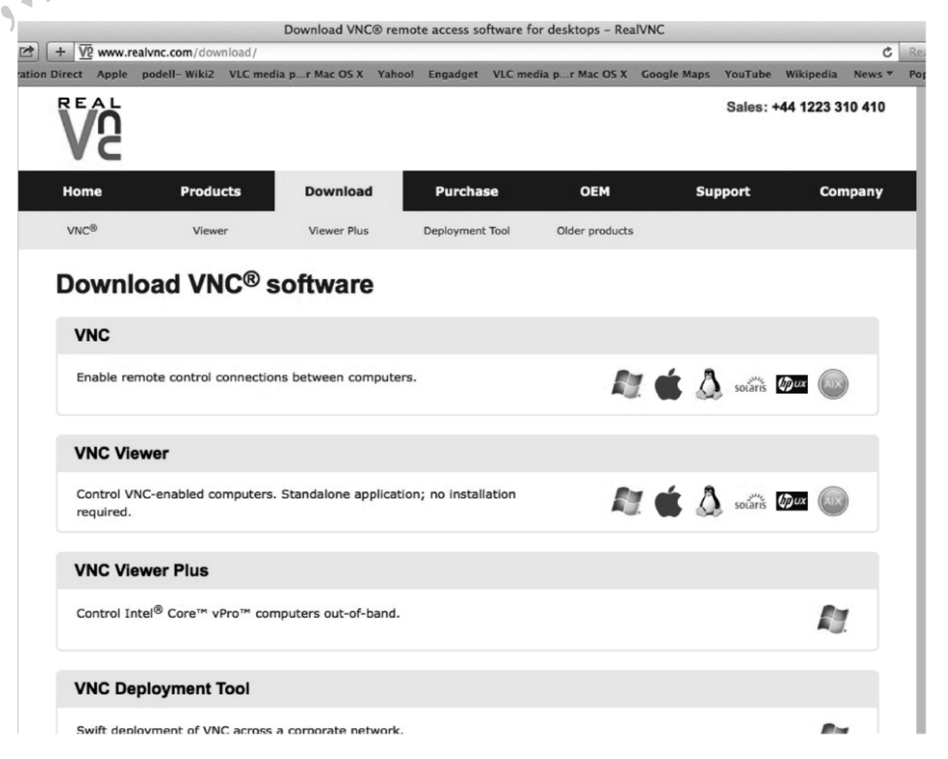


图 3-12　下载 VNC Viewer

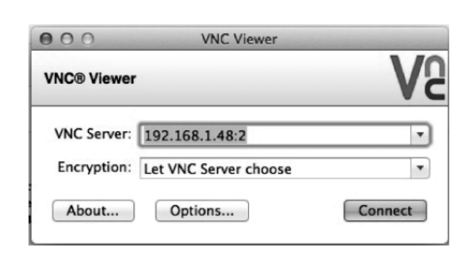


图 3-13　VNC View 连接设置

3.5　在 Android 手机使用 VNC Client 远程控制树莓派

案例 10

在 Android 设备上也可以使用 VNC Viewer，如图 3-14 所示，在手机上操作 Raspberry。Android 版本使用起来界面如图 3-15 所示。

图 3-14　在 Google Play 上的 VNC Viewer



图 3-15　在 Android 手机上操作树莓派

3.6 在 iOS 上使用 VNC Client 远程控制树莓派

案例 11

VNC 也有 iOS 设备的免费版本。只要在 Apple Store 上查找 VNC Viewer,就可以下载和安装,如图 3-16 所示。



图 3-16　在 Apple Store 上的 VNC Viewer

iOS 版本使用起来界面如图 3-17 所示。



图 3-17　VNC Viewer 在 iOS 上运行

3.7 自启动 VNC Server

案例 12

1. 介绍

实际上执行到这里就完成了,但是每次开机都要这样重新执行一次。如果要设置为每次引导树莓派,都会自动地引导 VNC Server,在下面步骤中加上一条打开时自动引导 VNC server 的命令,方便以后使用。

2. 步骤

STEP1:编辑文件

请执行如下命令,转换为管理器。

```
$ sudo bash
```

然后,通过文本编辑器设置自动引导的执行动作。

```
$ sudo nano /etc/init.d/vncboot
```

在编辑文本模式下输入下面的数据。

```
#! /bin/sh
# /etc/init.d/vncboot
USER = root HOME = /root
export USER HOME
echo "Starting Tight VNC Server"
/usr/bin/tightvncserver :1 – geometry 800x600 – depth 16 – pixelformat rgb565
exit 0
```

完成后,通过 Ctrl+O 组合键存储文件和 Ctrl+X 组合键离开文本编辑器,如图 3-18 所示。

STEP2:修改文件权限

通过以下命令,修改文件的权限。

```
# sudo chmod 755 /etc/init.d/vncboot
```

STEP3:更新引导配置文件

通过以下命令,告诉系统指定这个配置文件,这样系统引导时就会自动执行该文件。

```
# sudo update – rc.d vncboot defaults
```

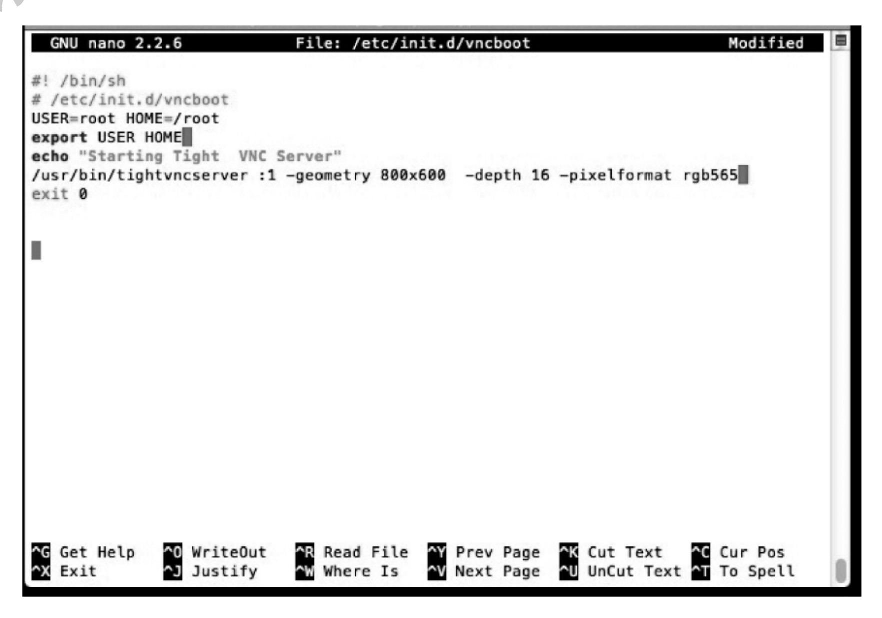


图 3-18 vncboot 文件的内容

然后,使用 reboot 命令重新引导树莓派,如图 3-19 所示,引导后就可以测试 VNC Server 是否可以正常工作。

```
# reboot
```

```
root@raspberrypi:/home/pi# sudo update-rc.d vncboot defaults
update-rc.d: using dependency based boot sequencing
insserv: warning: script 'K01vncboot' missing LSB tags and overrides
insserv: warning: script 'vncboot' missing LSB tags and overrides
insserv: script vncboot2: service vncboot already provided!
root@raspberrypi:/home/pi# sudo update-rc.d vncboot defaults
update-rc.d: using dependency based boot sequencing
insserv: warning: script 'K01vncboot' missing LSB tags and overrides
insserv: warning: script 'vncboot' missing LSB tags and overrides
insserv: script vncboot2: service vncboot already provided!
root@raspberrypi:/home/pi# reboot
```

图 3-19 更新引导配置文件

第 4 章

搭建网站服务器系统

4.1 搭建 PHP 网站服务器

案例 13

1. 介绍

Raspberry 用来搭建网站服务器是最棒的,它的体积小、额定功率小,只要有高网速就可以有很棒的效果。如果觉得 CPU 运行速度慢,本书第 23 章介绍了用多台树莓派来一起处理。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络连接设备。

需要有以下软件:

(1) Apache2;

(2) PHP;

(3) SSL。

3. 接线

请确认网络功能正常。

4. 步骤

推荐先设置好固定 IP,虽然使用动态 IP 也可以搭建网站,但是通过浏览器连接时,因为有固定 IP 地址,使用会比较方便。

STEP1:扩充 SD 卡

如果之前没有扩充 SD 卡,此时 SD 卡不论是 16GB 的还是 32GB 的,都只有不到 1GB 的空间,这因为当初在刻录时,使用了 img 格式,所以继承了原本的尺寸空间。使用以下指令来扩充 SD 卡的存储空间:

```
$ sudo raspi-config
```

选中 1 Select Expand Filesystem 之后,单击 Finish 按钮,如图 4-1 所示。

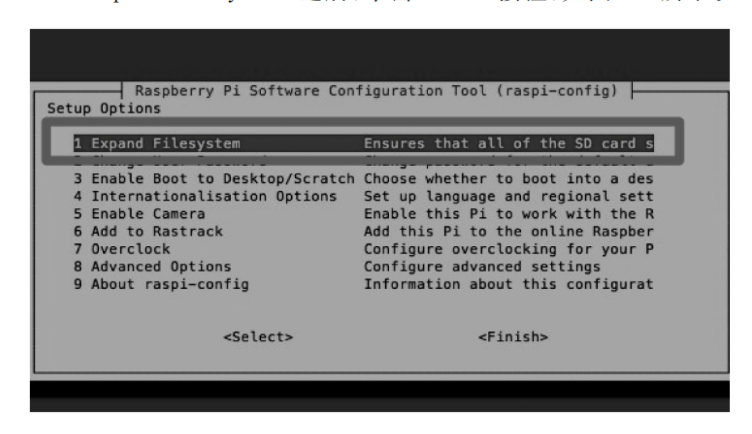


图 4-1 扩大 SD 卡的存储空间

STEP2:创建新的用户账号

为了信息安全起见,推荐使用以下命令创建新的用户账号,如图 4-2 所示,提供给专门从网络登录的用户使用。

```
$ sudo groupadd www - data
$ sudo usermod  - a  - G www - data www - data
```

```
pi@raspberrypi ~ $ sudo groupadd www-data
groupadd: group 'www-data' already exists
pi@raspberrypi ~ $ sudo usermod -a -G www-data www-data
```

图 4-2 创建新的用户账号

STEP3:搭建网络 Apache PHP

以下命令是下载安装 Apache2 和设置 PHP5,如图 4-3 所示。

```
$ sudo apt - get install apache2 php5 php5 - json php5 - gd php5 - sqlite curl libcurl3 libcurl4
- openssl - dev
$ sudo apt - get install php5 - curl php5 - gd php5 - cgi php - pear php5 - dev build - essential
$ sudo apt - get install libpcre3 - dev php5 libapache2 - mod - php5 php - apc gparted
```

```
pi@raspberrypi ~ $ sudo apt-get install apache2 php5 php5-json php5-gd php5-sqli
te curl libcurl3 libcurl4-openssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'php5-common' instead of 'php5-json'
apache2 is already the newest version.
curl is already the newest version.
libcurl3 is already the newest version.
libcurl4-openssl-dev is already the newest version.
```

图 4-3 安装 Apache2 和设置 PHP5

STEP4：设置 APC

通过以下命令安装 APC(Alternative PHP Cache 可选 PHP 缓存)，在安装过程中，会询问很多问题，只要选择默认值，一直按下 Enter 键即可，如图 4-4 所示。

```
$ sudo pecl install apc
```

```
pi@raspberrypi ~ $ sudo pecl install apc
downloading APC-3.1.13.tgz ...
Starting to download APC-3.1.13.tgz (171,591 bytes)
.....................................done: 171,591 bytes
55 source files, building
running: phpize
Configuring for:
PHP Api Version:         20100412
Zend Module Api No:      20100525
Zend Extension Api No:   220100525
Enable internal debugging in APC [no] :
Enable per request file info about files used from the APC cache [no] :
Enable spin locks (EXPERIMENTAL) [no] :
Enable memory protection (EXPERIMENTAL) [no] :
Enable pthread mutexes (default) [no] :
Enable pthread read/write locks (EXPERIMENTAL) [yes] :
building in /tmp/pear/temp/pear-build-rootkAggEZ/APC-3.1.13
```

图 4-4　安装 APC

会询问的问题有

(1) Enable internal debugging in APC (是否打开内部调试模式)：请选择 no(不要)。

(2) Enable per request file info about files used from the APC cache(引导每个请求的文件信息都使用 APC 缓存)：请选择 no(不要)。

(3) Enable memory protection(启用内存保护)(EXPERIMENTAL)[no]：请选 no(不要)。

(4) Enable pthread mutexes(启用线程互斥)(default)[no]：请选择 no(不要)。

(5) Enable pthread read/write locks(启用线程的读/写锁)(EXPERIMENTAL)[yes]：请选择 yes(是)。

然后，打开文本编辑器，修改 apc.ini 文件的数据。

```
$ sudo nano /etc/php5/cgi/conf.d/apc.ini
```

添加以下设置，如图 4-5 所示。

```
extension = apc.so
apc.enabled = 1
apc.shm_size = 30
```

通过 Ctrl+O 组合键存储后，再通过 Ctrl+X 组合键离开 nano 文本编辑器。

STEP5：修改网页设置 PHP

修改 PHP 配置文件，来调整可以上传的文件尺寸。

```
$ sudo nano /etc/php5/apache2/php.ini
```



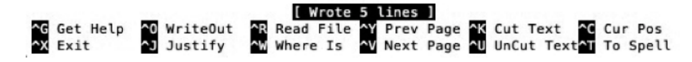


图 4-5　配置 apc. ini

在 nano 文本编辑器中，通过 Ctrl＋W 组合键来寻找 upload_max_filesize 的参数，把它改大一点，因为此参数会影响到时候可以上传的文件尺寸。如下改到 1GB：

```
upload_max_filesize = 1GB
post_max_size = 1GB
```

然后在这个文件中，找到 extension＝，添加如图 4-6 所示的值。

```
extension = apc.so
```

```
powenko — pi@raspberrypi: ~ — ssh — 80×24
GNU nano 2.2.6          File: /etc/php5/apache2/php.ini          Modified

;auto_detect_line_endings = Off

;;;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;;

extension=apc.so

; If you wish to have an extension loaded automatically, use the following
; syntax:
;
;   extension=modulename.extension
;
; For example, on Windows:
;
;   extension=msql.dll
;
; ... or under UNIX:
;
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 4-6　修改网页设置 php. ini

STEP6：配置文件权限

为了方便起见，通过以下命令配置文件的访问权限。

```
$ sudo nano /etc/apache2/sites-enabled/000-default
```

把文本里的 none 改为 all，如图 4-7 所示。

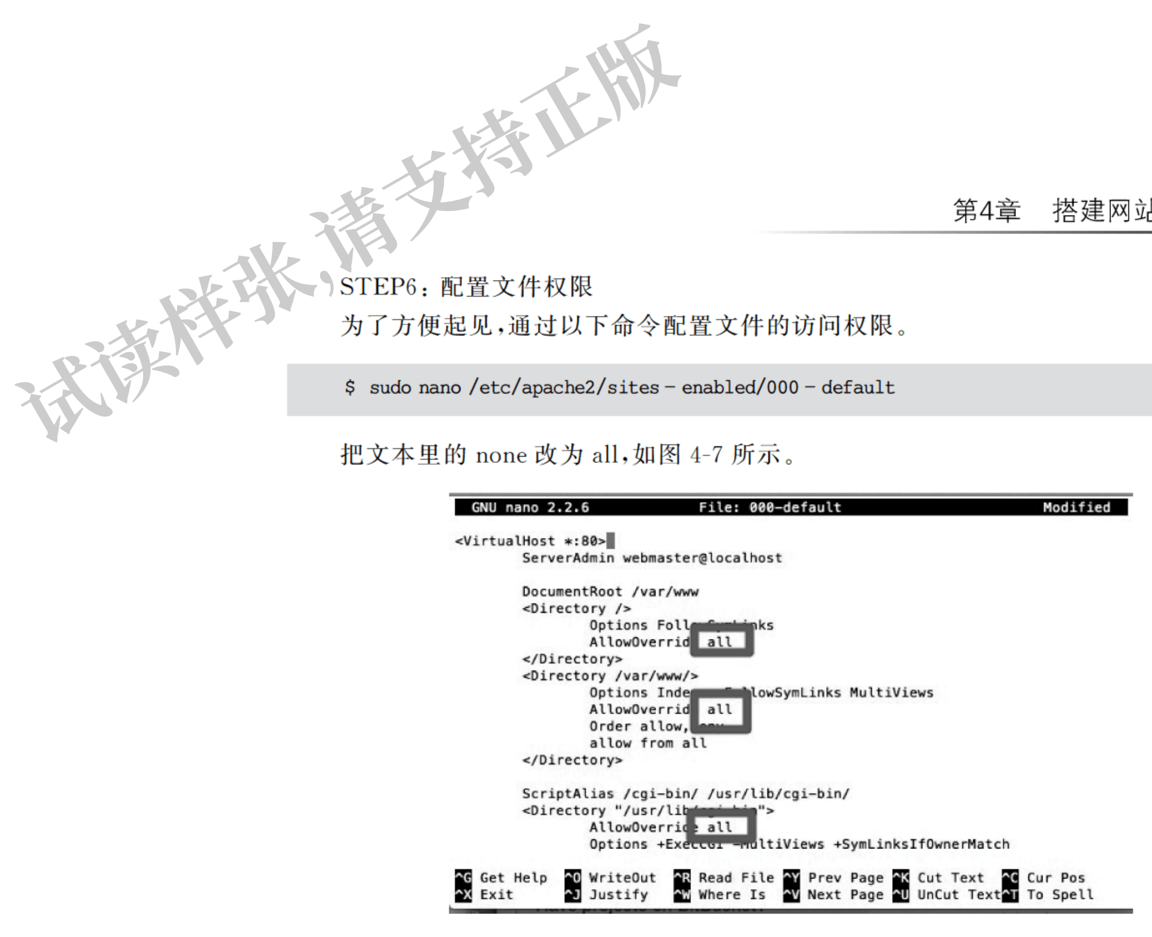


图 4-7　配置 SSL

通过 Ctrl+O 组合键存储后，再通过 Ctrl+X 组合键离开 nano 文本编辑器。

STEP7：重新引导 Apache2

通过以下命令重新引导 Apache2，如图 4-8 所示，这样 Apache2 会去引导刚刚所设置的 SSL 安全配置。

```
$ sudo service apache2 restart
$ sudo a2enmod rewrite
$ sudo a2enmod headers
```

```
pi@raspberrypi /etc/apache2/sites-enabled $ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  service apache2 restart
pi@raspberrypi /etc/apache2/sites-enabled $ sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  service apache2 restart
```

图 4-8　重新引导 Apache2

STEP8：设置 SSL 的密码

设置 SSL 的相关数据和密码，通过如图 4-9 所示的命令进行配置。

```
$ sudo openssl genrsa - des3 - out server.key 1024; sudo openssl rsa - in server.key - out server.key.insecure; sudo openssl req - new - key server.key - out server.csr; sudo openssl x509 - req - days 365 - in server.csr - signkey server.key - out server.crt; sudo cp server.crt /etc/ssl/certs; sudo cp server.key /etc/ssl/private; sudo a2enmod ssl; sudo a2ensite default - ssl
```

在过程中,会要求输入密码、国家别名、公司名称、个人名称等认证数据。

Enter pass phrase for server.key:输入密码。

Verifying-Enter pass phrase for server.key:再次输入密码。

```
pi@raspberrypi ~ $ sudo openssl genrsa -des3 -out server.key 1024; sudo openssl rsa -in server.key -out server.key.insecure;sudo openssl req -new -key server.key -out server.csr;sudo openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt;sudo cp server.crt /etc/ssl/certs;sudo cp server.key /etc/ssl/private;sudo a2enmod ssl;sudo a2ensite default-ssl
Generating RSA private key, 1024 bit long modulus
.......................++++++
.++++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
Enter pass phrase for server.key:
writing RSA key
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
```

图 4-9 设置 openSSL 密码

STEP9:引导 PHP Apache2

完成之后,通过以下命令重新引导,就可以使用 SSL 安全机制了。

```
$ sudo service apache2 restart
```

STEP10:测试网页

通过以下查看现在的网络 IP 地址:

```
$ ifconfig
```

确定 IP 之后,就可以在浏览器输入 IP 地址查看网站是否架设成功,如图 4-10 所示。

WWW 文件位于/var/www 路径下,可以把 PHP 程序或 HTML 文件放在这个路径中,网站就可以工作了。

5. 结果

通过浏览器来确认网站可以正常工作,如图 4-11 所示。

6. 教学影片

完整的教学视频可以看 raspberrypi_00apache2。

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:76:58:3e
          inet addr:192.168.0.106  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13192 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6424 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10101256 (9.6 MiB)  TX bytes:1474752 (1.4 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:46 errors:0 dropped:0 overruns:0 frame:0
          TX packets:46 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4046 (3.9 KiB)  TX bytes:4046 (3.9 KiB)
```

图 4-10　使用 ifconfig 命令查看 IP 位置

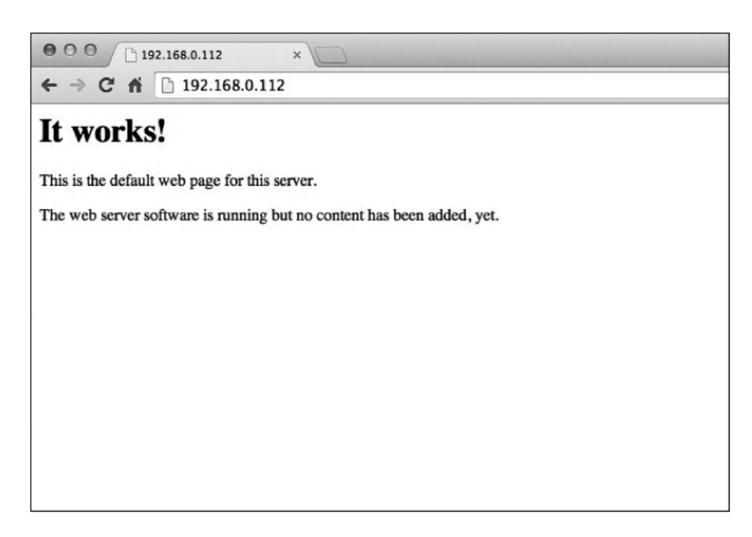


图 4-11　浏览器输入 IP 地址

4.2　运行 PHP 程序

案例 14

1．介绍

介绍如何在树莓派上编写简单的 PHP 程序，请先按照 4.1 节，搭建 PHP 的网站，只要在/var/www 之中存放网页程序，就可以显示和使用了。在此案例中，通过文字编辑器来写个简单的 PHP 程序。

2．步骤

编写 PHP 文件。

```
$ sudo nano /var/www/my.php
```

在文本编辑器中输入以下程序：

```
1.  <h1>Hello PHP,www.powenko.com</h1>
2.  <?php
3.          phpinfo();
4.  ?>
```

程序解说

第 1 行：使用标准的 HTML 语言显示一段文字 Hello PHP，www.powenko.com。

第 2 行和第 4 行：<?php ?>在这个范围内的文字是程序语言，并且是 PHP 程序语言。

第 3 行：PHP 程序语言函数 phpinfo()；用来显示树莓派上 PHP 程序语言的相关数据。

按下键盘上的 Ctrl+O 组合键做存储，再按下键盘上的 Ctrl+X 组合键离开 nano 文本器。

3. 测试 PHP 程序

在浏览器中输入 IP 地址并且加上/my.php 后缀，例如 http://192.169.1.10/my.php，如图 4-12 所示就能看到执行的结果。

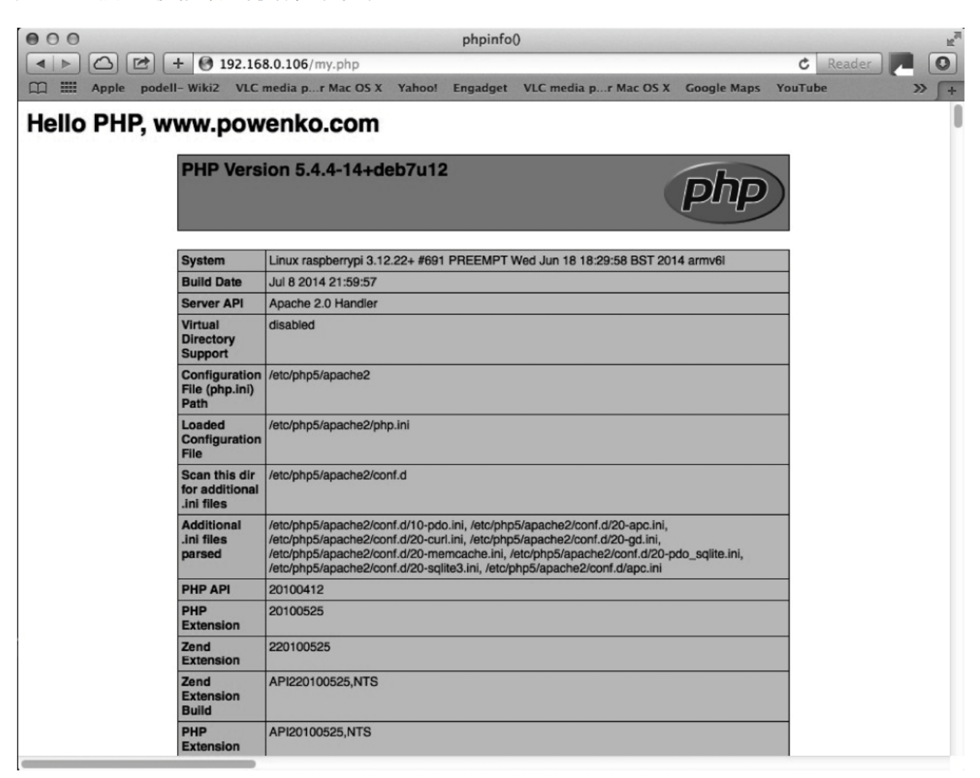


图 4-12　测试 PHP 程序的执行结果

4. 教学影片

完整的教学视频可以看 RaspberryPi_PHP_my_php_T。

4.3　创建数据库 MySQL

案例 15

1. 介绍

MySQL 是世界上最受欢迎的开源数据库,也是关系型数据库管理系统,而且是当前市面上最快速和最稳定的数据库之一。当前搭建的商业网站中,PHP 加上 MySQL 数据库使用最为广泛,就像是微软的. NET 搭配 SQL Server 一样。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络连接设备。

需要有以下软件:

(1) Apache2;

(2) PHP;

(3) SSL。

3. 接线

请确认网络功能正常。

4. 步骤

首先完成“4.1 搭建 PHP 网站服务器”。

STEP1: 安装 MySQL 数据库

输入以下命令把 PHP5 程序语言中的 libapache2-mod-php5 和 php5-mysql 模块安装到树莓派中,如图 4-13 所示。

```
$ sudo apt-get install mysql-server mysql-client php5-mysql
```

php5-mysql 模块提供 PHP 连接 MySQL 数据库的相关函数与模块。

```
pi@raspberrypi ~ $ sudo apt-get install mysql-server mysql-client php5-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
php5-mysql is already the newest version.
The following extra packages will be installed:
  heirloom-mailx libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl
  libmysqlclient16 libnet-daemon-perl libplrpc-perl mysql-client-5.5 mysql-server-5.5
  mysql-server-core-5.5
Suggested packages:
```

图 4-13　安装 MySQL 和 PHP5 程序语言和模块

STEP2：设置 MySQL 的账号和密码

安装的过程中，会询问 MySQL 管理器 root 账号的密码，如图 4-14 所示，按照下面的指示输入密码并记录下来，以后每次登录 MySQL 都会询问该密码。

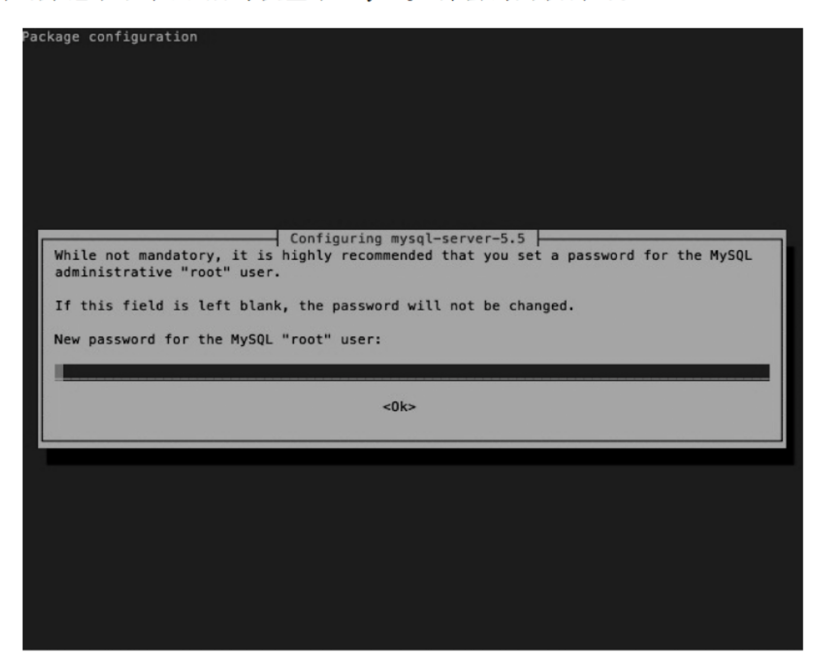


图 4-14　输入 MySQL 管理器权限的密码

重新输入一次相同的密码做确认。

STEP3：下载安装 MySQL 在 PHP5 上的模块

输入以下命令来安装 MySQL 在 PHP5 上的模块，这个模块主要让管理器通过网页来管理数据库。

```
$ sudo apt-get install phpmyadmin
```

安装过程中，phpMyAdmin 安装网页服务器管理软件，和询问密码，把刚才设置的密码再输入一次就可以了。

接下来修改 Apache2 的设置。

```
$ sudo nano /etc/apache2/apache2.conf
```

在文件的最后加上以下配置，如图 4-15 所示。

```
Include /etc/phpmyadmin/apache.conf
```

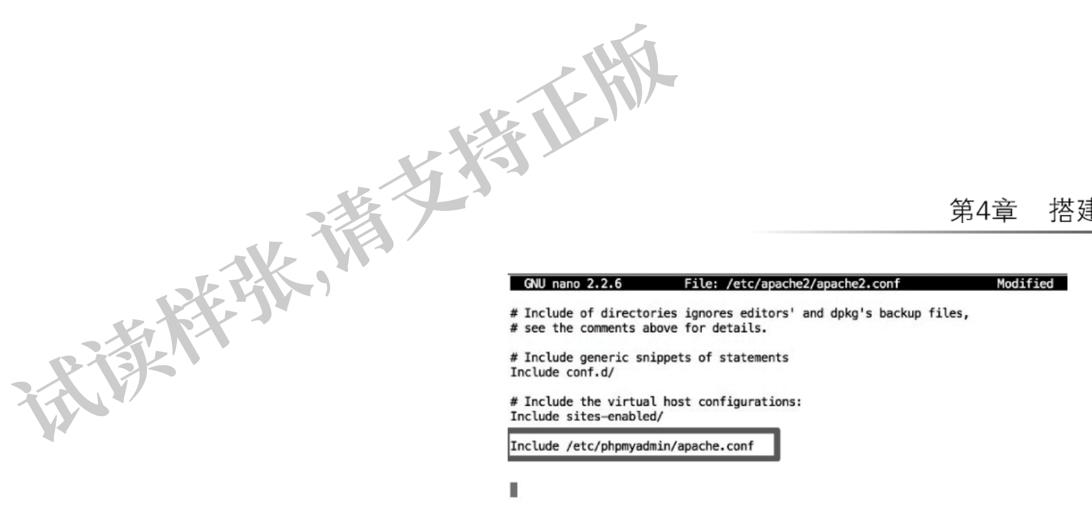


图 4-15　修改 Apache2 的配置

存储后，需要重新引导网站服务器 Apache。

```
$ sudo /etc/init.d/apache2 restart
```

STEP4：设置 phpMyAdmin

打开浏览器输入网址和\phpmyadmin 的路径，就可以进入 phpMyAdmin，如图 4-16 所示。

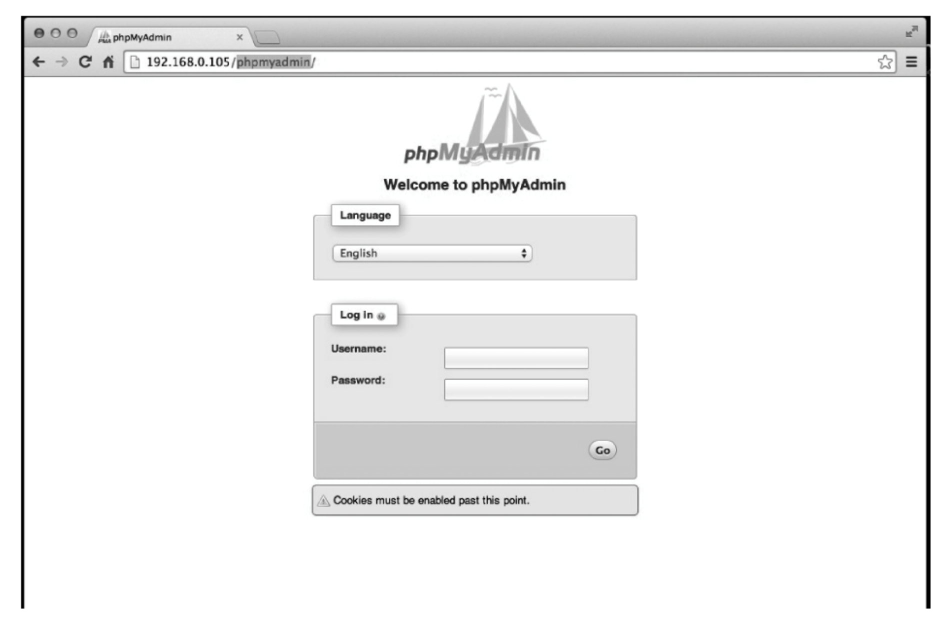


图 4-16　通过浏览器登录并打开账号 root 和密码

例如 http://192.168.0.111/phpmyadmin，账号请使用 root，密码就是在安装时设置的。

注意 如果不小心把账号和密码忘记了,可以输入以下命令进行重置。

```
$ mysqladmin -u root -p'oldpassword' password 'newpass'
```

例如

```
mysqladmin -u root -p'abc' password '123456'
```

如果顺利就会看到如图 4-17 的界面。

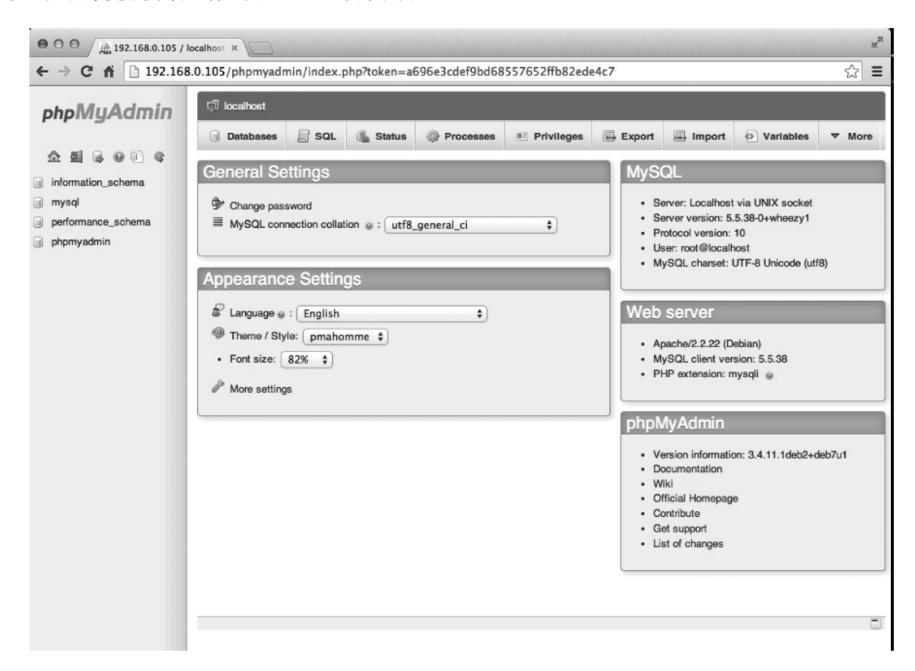


图 4-17 进入 phpMyAdmin 设置数据库

5. 教学影片

完整的教学视频可以看 raspberryPi_MySQL。

4.4 搭建博客——WordPress

案例 16

1. 介绍

世界上搭建博客的人当中,使用 WordPress 是最多的,作者也是 WordPress 爱用者,如图 4-18 所示,作者的个人网站也是使用这套系统搭建的。有相当多的书籍和资料,介绍如

何设置和美化它，最厉害的是它有一大堆外挂程序和外观式样可以供用户使用，并且很多都是完全免费的。



图 4-18　作者使用 WordPress 搭建的个人网站 www.powenko.com

2. 设备

需要有以下硬件：

（1）树莓派开发板；

（2）网络连接设备。

需要有以下软件：

（1）Apache2；

（2）PHP；

（3）SSL；

（4）MySQL。

3. 接线

请确认网络功能正常。

4. 步骤

先完成“4.1 搭建 PHP 网站服务器”和“4.3 创建 MySQL 数据库”。

STEP1：移动到树莓派的网页路径

输入以下命令将工作路径移动到树莓派的网页路径/var/www，如图 4-19 所示。

```
pi@raspberrypi ~ $ cd /var/www
pi@raspberrypi /var/www $ ls
index.html
pi@raspberrypi /var/www $
```

图 4-19　移动到/var/www 路径

```
$ cd /var/www
```

STEP2：下载 WordPress

可以通过浏览器到官方网站找到最新版本的下载路径，然后复制下载路径。或者直接输入以下命令就可以取得最新的版本，如图 4-20 所示。

```
$ sudo wget https://wordpress.org/latest.zip
```

```
pi@raspberrypi /var/www/wordpress $ sudo wget https://wordpress.org/latest.zip
--2014-08-13 04:04:43--  https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org)... 66.155.40.250, 66.155.40.249
Connecting to wordpress.org (wordpress.org)|66.155.40.250|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6341903 (6.0M) [application/zip]
Saving to: `latest.zip'

100%[======================================>] 6,341,903   2.06M/s   in 2.9s

2014-08-13 04:04:50 (2.06 MB/s) - `latest.zip' saved [6341903/6341903]
```

图 4-20 通过 wget 下载 WordPress

STEP3：解压缩文件

使用 unzip 命令解压缩文件，如图 4-21 所示。

```
$ sudo unzip latest.zip
```

```
pi@raspberrypi /var/www/wordpress $ sudo unzip  latest.zip
Archive:  latest.zip
   creating: wordpress/
  inflating: wordpress/wp-settings.php
  inflating: wordpress/wp-cron.php
  inflating: wordpress/wp-comments-post.php
  inflating: wordpress/wp-activate.php
   creating: wordpress/wp-admin/
  inflating: wordpress/wp-admin/link-parse-opml.php
```

图 4-21 解压缩文件

STEP4：确保只有一个 index 文件

使用 ls 命令查看文件，确保只有一个 index 文件。如果有 index.htm 或 index.html 都请使用 mv 命令修改文件名，如图 4-22 所示。例如把 index.html 修改成 i2.html。

```
$ mv index.html i2.html
```

并且把下载的 zip 文件删除。

```
$ rm wordpress.zip
```

STEP5：创建数据库

创建一个全新的数据库给 WordPress 使用，打开浏览器输入实际 IP 地址添加后缀

```
pi@raspberrypi /var/www/wordpress $ ls
index.html       wp-admin             wp-includes         wp-signup.php
index.php        wp-blog-header.php   wp-links-opml.php   wp-trackback.php
latest.zip       wp-comments-post.php wp-load.php         xmlrpc.php
license.txt      wp-config-sample.php wp-login.php
readme.html      wp-content           wp-mail.php
wp-activate.php  wp-cron.php          wp-settings.php
pi@raspberrypi /var/www/wordpress $ mv index.html i2.html
mv: cannot move `index.html' to `i2.html': Permission denied
pi@raspberrypi /var/www/wordpress $ sudo mv index.html i2.html
pi@raspberrypi /var/www/wordpress $ ls
i2.html          wp-admin             wp-includes         wp-signup.php
index.php        wp-blog-header.php   wp-links-opml.php   wp-trackback.php
latest.zip       wp-comments-post.php wp-load.php         xmlrpc.php
license.txt      wp-config-sample.php wp-login.php
readme.html      wp-content           wp-mail.php
wp-activate.php  wp-cron.php          wp-settings.php
```

图 4-22　博客的相关文件放在/var/www/wordpress

/phpmyadmin，进入 MySQL 数据库管理系统。按照图 4-23 所示的方法完成设置。

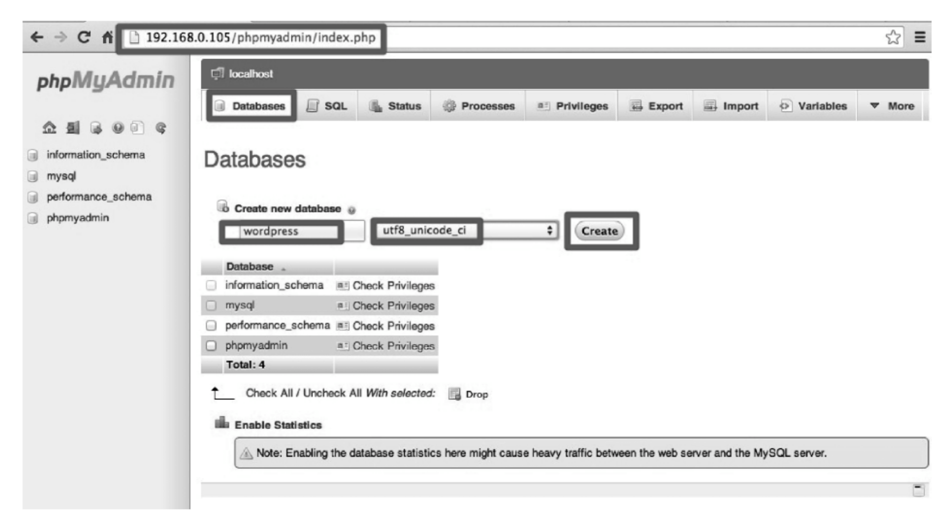


图 4-23　创建一个新的数据库

(1) 单击 Databases 进入数据库管理系统。

(2) 在 Create new database 中定义数据库名称为 wordpress。

(3) 在下拉列表中选择数据库的数据编码，这里使用的是 utf8_unicode_ci。

(4) 最后，单击 Create 按钮，就可以创建一个新全新的数据库。

STEP6：设置 WordPress 博客

在浏览器中输入实际的 IP/wordpress 连接到 WordPress 博客上，并且把相关的信息填入，如图 4-24 所示。

(1) Database Name 数据库名称：wordpress。

(2) User Name 数据库的用户名称：root。

(3) Password 数据库的密码：输入设置的密码。

(4) Table Prefix 数据库的表添加前缀：wp_。

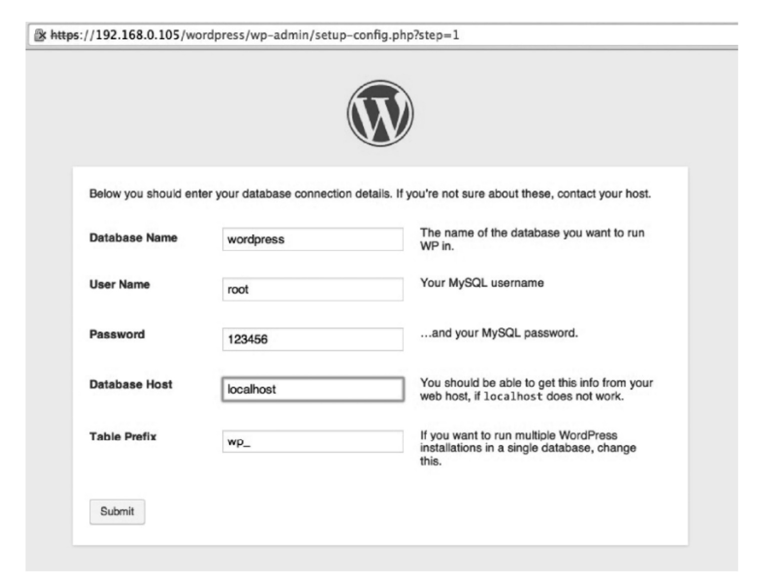


图 4-24 设置 wordpress 博客

(5) 确认之后单击 Sumit 按钮。

STEP7：设置 wp-config. php 内容

设置完博客后，如图 4-25 所示，下一个是复制界面中阴影部分，创建一个 wp-config. php 文件，把内容粘贴上。




```
<?php
/**
 * The base configurations of the WordPress.
 *
 * This file has the following configurations: MySQL settings, Table Prefix,
 * Secret Keys, WordPress Language, and ABSPATH. You can find more information
 * by visiting {@link http://codex.wordpress.org/Editing_wp-config.php Editing
 * wp-config.php} Codex page. You can get the MySQL settings from your web host.
 *
 * This file is used by the wp-config.php creation script during the
 * installation. You don't have to use the web site, you can just copy this file
 * to "wp-config.php" and fill in the values.
 *
 * @package WordPress
 */
```

图 4-25 wp-config. php 内容

输入以下命令，进入文本编辑器，记得将 wp-config. php 文件放在 WordPress 同一个路径中，如图 4-26 所示。

```
$ sudo nano wp－config.php
```

```
pi@raspberrypi /var/www/wordpress $ sudo nano wp-config.php
```

图 4-26　编辑 wp-config. php 文件

在 nano 文本编辑器中，使用键盘上的 Ctrl＋V 组合键将内容粘贴，并且使用 Ctrl＋O 组合键存储数据，Ctrl＋X 组合键离开文本编辑器，如图 4-27 所示。

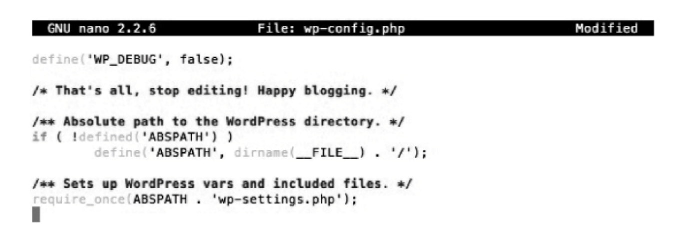


图 4-27　编辑 wp-config. php 文件

STEP8：设置 WordPress 博客的标题

在浏览器中再次输入实际的 IP/wordpress 地址，显示信息就不一样了，有关博客的信息填写，如图 4-28 所示。

Site Title 博客的标题。

Username 博客的管理员账号：admin。

Password 博客的管理员密码：输入设置的密码。

Password，twice 再一次输入密码。

Your email 你的 Email 地址。

Privacy 保密性：勾选的话，就可以搜索到这个网站。

最后单击 install wordpress 按钮就完成。

5. 结果

完成的效果如图 4-29 所示。

设置完成后，通过博客的管理账号登录后台管理，然后可以添加文章、式样等功能，如图 4-30 所示。

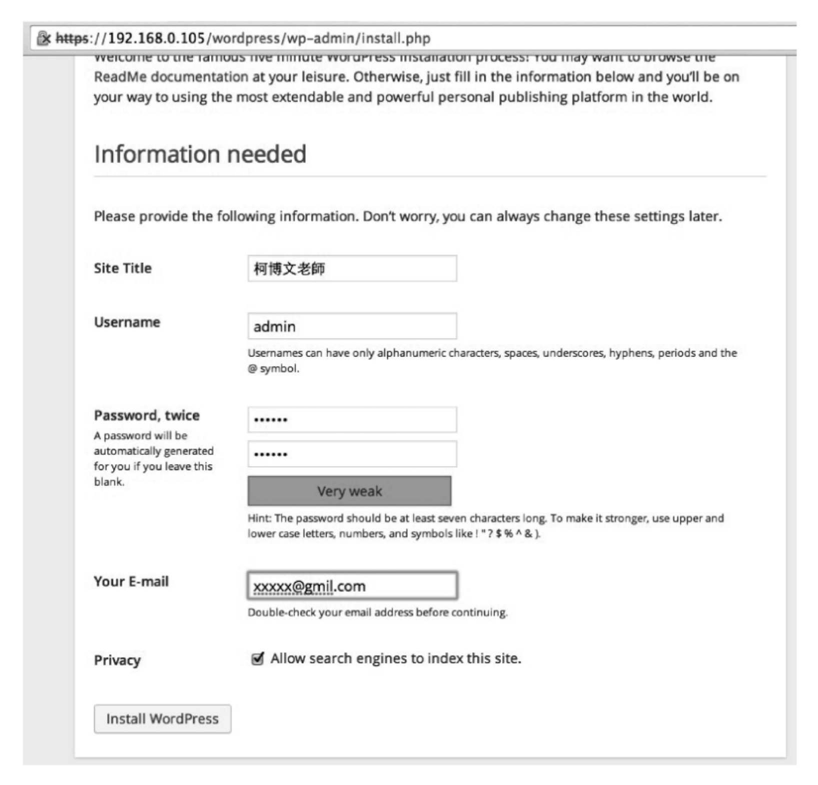


图 4-28　设置 WordPress 博客的标题

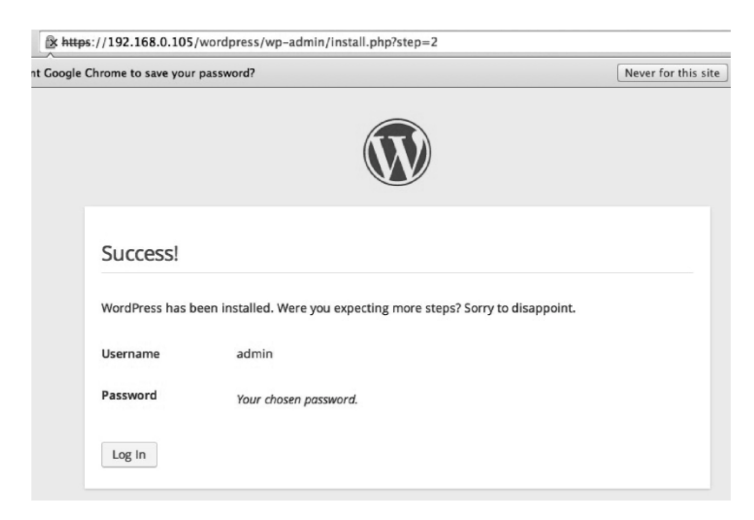


图 4-29　完成设置



图 4-30　WordPress 管理后台

用户使用网页登录,可以看到 WordPress 的博客网站,如图 4-31 所示,可以自由地修改添加,当前已有支持中文字库的版本,可以进行汉化。

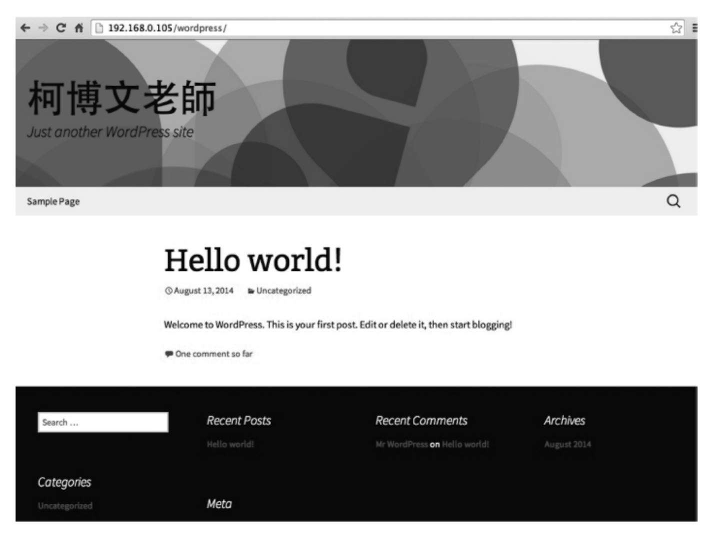


图 4-31　个人博客搭建完成后的效果

完成后的展示效果,可以参考视频 raspberryPi_Wordpress_demo。

6. 教学影片

完整的教学视频可以看 raspberryPi_Wordpress。

第 5 章 网络附加存储 NAS 系统与 USB 外接硬盘文件服务器

本章将会把树莓派搭建为文件服务器,让家中的手机、平板电脑、台式计算机、笔记本电脑通过网络附加存储系统共享与备份重要数据。

网络附加存储(Network Attached Storage,NAS)是一种专门的数据存储技术的名称,它可以直接连接在计算机网络上,对网络用户提供集中式数据访问服务。

5.1 添加 USB 外接硬盘

案例 17

1. 介绍

一般情况下,如果 SD 卡的容量不足,树莓派可以使用 USB 外接硬盘。按照本节的步骤,可以让家中的 USB 外接硬盘提供给树莓派使用。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) USB 外接硬盘。

需要有以下软件:

TightVNC Server 软件。

推荐的 USB 外接硬盘,是使用外接电源的外接式硬盘,而不是单单靠 USB 供电的外接硬盘,因为树莓派本身的功率不足以推动外接式硬盘。

3. 接线

把外接硬盘插到 USB 接口,如图 5-1 所示。

4. 步骤

STEP1:备份 USB 硬盘数据

输入以下命令更新安装软件:

```
$ sudo apt - get update
$ sudo apt - get upgrade
```



图 5-1　USB 外接硬盘插到树莓派上

STEP2：确认树莓派识别该硬盘

输入以下命令确认树莓派已经识别该硬盘，可以通过硬盘名称和硬盘容量来确认，如图 5-2 所示。

```
$ sudo fdisk -l
```

```
pi@raspberrypi /dev $ sudo fdisk -l

Disk /dev/mmcblk0: 8068 MB, 8068792320 bytes
4 heads, 16 sectors/track, 246240 cylinders, total 15759360 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000c4661

        Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1            8192      122879       57344    c  W95 FAT32 (LBA)
/dev/mmcblk0p2          122880    15759359     7818240   83  Linux

Disk /dev/sda: 250.1 GB, 250059350016 bytes
255 heads, 63 sectors/track, 30401 cylinders, total 488397168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               2   488397167   244198583    b  W95 FAT32
```

图 5-2　硬盘情况

如果没有看到硬盘,可以通过以下方法连接:

(1) 重启树莓派。

(2) 使用带外接电源的硬盘,不要只用树莓派上的 USB 供电,功率会不足。

(3) 不要使用 USB Hub。

(4) 更换外接硬盘盒,作者曾遇到树莓派不能读取 USB 3.0 接口,更换 USB 2.0 接口的外接硬盘盒就识别了。

STEP3:安装执行 NTFS 硬盘软件

输入以下命令安装 ntfs-3g:

```
$ sudo apt-get install ntfs-3g
```

STEP4:指定 USB 硬盘的位置

第一次挂载 USB 外接硬盘,需要创建一个路径,例如名称 usbhdd 可以自己命名。

```
$ sudo mkdir /media/usbhdd
```

并且指定该路径的用户权限。

```
$ sudo chown pi:pi /media/usbhdd
```

指定//media/usbhdd 的路径,会挂载到硬件 dev/sda1 上面。

```
$ sudo mount -t vfat -o uid=pi,gid=pi /dev/sda1 /media/usbhdd
```

如果出现下图的错误消息,如图 5-3 所示,意味着文件格式有误。请确定硬盘是 FAT32 格式,如果要在树莓派上格式化硬盘,可以参考“5.2 格式化硬盘”。或者推荐在你熟悉的 Windows 操作系统中进行格式化。

```
pi@raspberrypi ~ $ sudo mount -t vfat -o uid=pi,gid=pi /dev/sda1 /media/usbhdd
mount: wrong fs type, bad option, bad superblock on /dev/sda1,
       missing codepage or helper program, or other error
       In some cases useful info is found in syslog - try
       dmesg | tail  or so
```

图 5-3 错误消息

STEP5:确认硬盘的数据

此时可以通过该路径,访问 USB 硬盘中的数据,如图 5-4 所示。

```
$ cd /media/usbhdd
$ ls
```

如果要离开或拔除 USB 硬盘,同样也要进行硬盘卸载。

```
pi@raspberrypi ~ $ sudo mkdir /media/usbhdd
pi@raspberrypi ~ $ sudo chown pi:pi /media/usbhdd
pi@raspberrypi ~ $ sudo mount -t vfat -o uid=pi,gid=pi /dev/sda1 /media/usbhdd
pi@raspberrypi ~ $ cd /media/usbhdd/
pi@raspberrypi /media/usbhdd $ ls
pi@raspberrypi /media/usbhdd $ nano 1.txt
pi@raspberrypi /media/usbhdd $ ls
1.txt
```

图 5-4 挂载硬盘

```
$ sudo umount /media/usbhdd
```

STEP6：设置开机自动挂载

每次开机都需要手动挂载一次，很麻烦。按照以下的方法，设置开机时自动挂载 USB 硬盘。首先备份 fstab 的硬盘配置文件。

```
$ sudo cp /etc/fstab /etc/fstab.backup
```

然后查看该外接式硬盘的 UUID，可以通过以下命令查询，如图 5-5 所示。

```
$ sudo blkid
```

```
pi@raspberrypi ~ $ sudo blkid
/dev/mmcblk0p1: SEC_TYPE="msdos" LABEL="boot" UUID="056E-4E83" TYPE="vfat"
/dev/mmcblk0p2: UUID="af599925-1134-4b6e-8883-fb6a99cd58f1" TYPE="ext4"
/dev/sda1: LABEL="HD256G" UUID="1599-18FA" TYPE="vfat"
```

图 5-5 查看 UUID 和 TYPE

以图 5-5 为例，会看到/dev/sda1：LABEL＝"HD256G" UUID＝"1599-18FA" TYPE＝"vfat"，重点是 UUID＝"1599-18FA" 和 TYPE＝"vfat"，按照实际情况进行以下修改。

```
$ sudo nano /etc/fstab
```

修改以下语句，填写 USB 硬盘的实际 UUDI 和 TYPE 值，如图 5-6 所示。

```
UUID = "1599 - 18FA"   /media/usbhdd   vfat      rw,defaults 0 0
```

注意 可以用 auto 代替 vfat 或 ext3 自动获取硬盘的格式，例如图 5-6 的设置可以改成：

```
UUID = "1599 - 18FA"   /media/usbhdd   auto      rw,defaults 0 0
```

STEP7：测试 mount

可输入以下命令测试/etc/fstab 文件是否有误，如果没有设置好就会出现如图 5-7 所示的信息。

```
$ sudo mount  - a
```

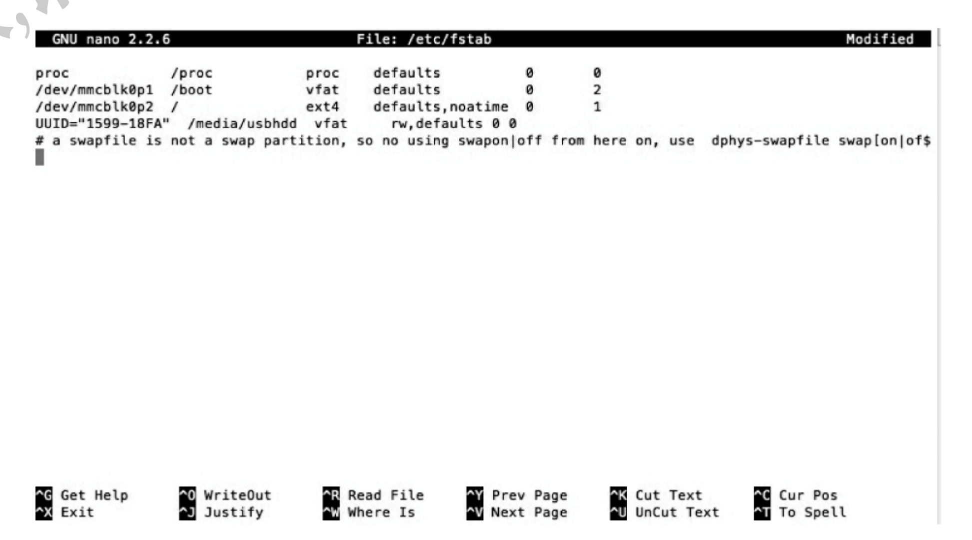


图 5-6　在/etc/fstab 文件配置外接式硬盘信息

```
pi@raspberrypi /media/usbhdd $ sudo mount -a
mount: mount point auto does not exist
```

图 5-7　mount 出现错误的消息

一切设置好后，重新开机就可以看到自动挂载的 USB 硬盘了。

```
$ sudo reboot
```

5. 结果

重新开机后，以后每次登录树莓派，就可以进入/media/usbhdd 访问外接硬盘，如图 5-8 所示。

```
$ cd /media/usbhdd
```

```
pi@raspberrypi ~ $ cd /media
pi@raspberrypi /media $ ls
usbhdd
pi@raspberrypi /media $ cd usbhdd
pi@raspberrypi /media/usbhdd $
```

图 5-8　访问外接硬盘

6. 教学影片

完整的教学视频可以看 raspberryPi_USBHD。

5.2　格式化硬盘

案例 18

1. 介绍

一般情况下,常用的 FAT32 格式在 Linux 系统中会有问题,此案例介绍如何把一个 FAT32 格式的 USB 外接式硬盘,格式化为 Linux 常用的 ext3 格式。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) USB 外接硬盘。

3. 接线

把 USB 外接硬盘连接到树莓派的 USB 接口上,推荐使用带外接电源的硬盘。

4. 步骤

STEP1:确认硬盘

可以找一个旧的硬盘,备份一下数据以备使用。大多数硬盘都是采用 Windows 操作系统的 NTFS,FAT32 等格式。而 Linux 的硬盘格式不太一样,需要把整个硬盘重新格式化。输入以下命令,确认树莓派识别现有的硬盘,如图 5-9 所示。

```
$ sudo fdisk －l | grep '^Disk'
```

```
pi@raspberrypi /dev $ sudo  fdisk -l | grep '^Disk'
Disk /dev/mmcblk0: 8068 MB, 8068792320 bytes
Disk identifier: 0x000c4661
Disk /dev/sda: 250.1 GB, 250059350016 bytes
Disk identifier: 0x00000000
```

图 5-9　确认硬盘

STEP2:确认硬盘格式

通过以下命令,确认树莓派已经识别该硬盘,可以通过硬盘名称和容量进行确认,确认硬盘 Device Boot,如图 5-10 所示,硬盘挂载在/dev/sda1 下。

```
$ sudo blkid
```

得知现在硬盘格式为 vfat,如果要让这个硬盘提供给网页 Apache 使用,需要转换成 ext3 格式。

```
pi@raspberrypi ~ $ sudo blkid
/dev/mmcblk0p1: SEC_TYPE="msdos" LABEL="boot" UUID="056E-4E83" TYPE="vfat"
/dev/mmcblk0p2: UUID="af599925-1134-4b6e-8883-fb6a99cd58f1" TYPE="ext4"
/dev/sda1: LABEL="HD256G" UUID="1599-18FA" TYPE="vfat"
```

图 5-10　确认硬盘 Device Boot

STEP3：卸载硬盘

接下来格式化硬盘格式为 ext3,确认硬盘数据已经备份,为了避免错误,把其他的硬盘暂时先拔离,并且通过以下的命令,确认要格式化的硬盘已经卸载,把/dev/sda1 调整为实际的机器的 Device Boot 名称。

```
$ sudo umount /dev/sda1
```

如果出现如图 5-11 所示的消息,说明这个硬盘正被其他软件使用,输入参数 l 来强制退出。

```
$ sudo umount -l /dev/sda1
```

```
pi@raspberrypi /media/usbhdd $ sudo umount /dev/sda1
umount: /media/usbhdd: device is busy.
        (In some cases useful info about processes that use
         the device is found by lsof(8) or fuser(1))
```

图 5-11 不能退出硬盘的消息

STEP4：格式化硬盘为 ext3

输入以下命令格式化硬盘(见图 5-12)：

```
$ sudo mkfs.ext3 /dev/sda1
```

```
pi@raspberrypi /dev $ sudo umount /dev/sda1
pi@raspberrypi /dev $ sudo  mkfs.ext3 /dev/sda1
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
15269888 inodes, 61049645 blocks
3052482 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=0
1864 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: 109/1864
```

图 5-12 格式化硬盘

5. 结果

完成之后,可以通过以下指令查看硬盘格式是否为 ext3,如图 5-13 所示。

```
$ sudo blkid
```

```
pi@raspberrypi /dev $  sudo blkid
/dev/mmcblk0p1: SEC_TYPE="msdos" LABEL="boot" UUID="056E-4E83" TYPE="vfat"
/dev/mmcblk0p2: UUID="af599925-1134-4b6e-8883-fb6a99cd58f1" TYPE="ext4"
/dev/sda1: UUID="de7a5c60-58a7-4f13-9eb2-2cc26478117d" SEC_TYPE="ext2" TYPE="ext3"
```

图 5-13 确认硬盘 Device Boot

6. 教学影片

完整的教学视频可以看 raspberryPi_FormatHD。

7. 扩展学习

参考“5.1 添加 USB 外接硬盘”,做开机识别和自动挂载,这里列出文件/etc/fstab 的设置,把 UUID 和硬盘格式修改为实际的参数。

```
UUID = "de7a5c60 - 58a7 - 4f13 - 9eb2 - 2cc26478117d"   /media/usbhdd   ext3   rw,defaults 0 0
```

5.3　搭建文件服务器 Samba

案例 19

1. 介绍

1991 年,澳洲 Andrew Tridgell 先生为了让 DOS 系统和 UNIX 系统可以共享文件,在 UNIX 系统中写了一套 SMB 协议的软件。经过一段时间的发展正式命名为 Samba。后来 Samba 就开始有了长足的发展,并且加入了许多有志之士。现在 Samba 已经成为 UNIX 世界中 SMB/CIFS 协议的代名词。Samba 最初发展的目的是为了沟通 Windows 与 UNIX Like 这两个不同的操作系统平台。使用 Windows 系统的计算机,可以通过“网络邻居”看到网络中的其他计算机,并通过共享文件夹来分享文件。在其他的操作系统中也有相关的软件可以设置。

在树莓派上搭建 Samba 后,可以让环境中的各种不同的操作系统共同分享文件。树莓派上安装完 Samba,就能成为一台不关机的文件服务器,局域网内的所有计算机、平板电脑、手机都可以使用这台服务器共享文件。还可以按照“23.3.2 搭建分布式存储服务 DFS——GlusterFS”做数据备份。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) USB 外接硬盘(选配)。

需要有以下软件:

Samba

3. 接线

把树莓派连接网络,并设置固定 IP 地址。

4. 步骤

STEP1: USB 硬盘数据

根据个人情况决定文件服务器是否要使用外接式硬盘,设置外接式硬盘的方法参考“5.1 添加 USB 外接硬盘”。

STEP2：安装所需软件

需要从网络下载最新版本的软件，所以先更新 apt-get，如图 5-14 所示。

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

```
Desktop  ocr_pi.png  python_games
pi@raspberrypi ~ $ sudo apt-get update
Hit http://repository.wolfram.com stable Release.gpg
Hit http://repository.wolfram.com stable Release
Get:1 http://archive.raspberrypi.org wheezy Release.gpg [490 B]
Get:2 http://raspberrypi.collabora.com wheezy Release.gpg [836 B
Get:3 http://mirrordirector.raspbian.org wheezy Release.gpg [490
Get:4 http://archive.raspberrypi.org wheezy Release [7,227 B]
Get:5 http://raspberrypi.collabora.com wheezy Release [5,009 B]
Hit http://repository.wolfram.com stable/non-free armhf Packages
```

图 5-14　更新 apt-get

STEP3：安装远程文件系统

下载 samba 软件，如图 5-15 所示。

```
$ sudo apt-get install samba
```

```
pi@raspberrypi ~ $ sudo apt-get install samba
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  tdb-tools
Suggested packages:
  openbsd-inetd inet-superserver smbldap-tools ldb-tools ctdb
The following NEW packages will be installed:
  samba tdb-tools
0 upgraded, 2 newly installed, 0 to remove and 16 not upgraded.
Need to get 3,327 kB of archives.
After this operation, 20.1 MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

图 5-15　安装 Samba

STEP4：创建用户

创建 Samba 用户的账户和权限，这里创建 powenko 这个账户，如图 5-16 所示。

```
$ sudo adduser powenko
```

```
root@raspberrypi:/home# $sudo adduser powenko
Adding user `powenko' ...
Adding new group `powenko' (1004) ...
Adding new user `powenko' (1001) with group `powenko' ...
Creating home directory `/home/powenko' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for powenko
Enter the new value, or press ENTER for the default
        Full Name []: Powen Ko
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []: www.powenko.com
Is the information correct? [Y/n] Y
```

图 5-16　使用 adduser 创建用户账号

STEP5：在 Sambar 添加用户

创建好最终用户账户后，把该用户的数据输送到 Samba 上，输入以下的命令即可，如果想创建多个用户，重复执行步骤 4，如图 5-17 所示。

```
$ sudo usermod -a -G sambashare powenko
$ sudo pdbedit -a -u powenko
```

```
root@raspberrypi:/home# sudo usermod -a -G sambashare powenko
root@raspberrypi:/home# sudo pdbedit -a -u powenko
new password:
retype new password:
Unix username:        powenko
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-355674155-2926933600-3750856622-1000
Primary Group SID:    S-1-5-21-355674155-2926933600-3750856622-513
Full Name:            Powen Ko,,,,www.powenko.com
Home Directory:       \\raspberrypi\powenko
HomeDir Drive:
Logon Script:
Profile Path:         \\raspberrypi\powenko\profile
Domain:               RASPBERRYPI
Account desc:
Workstations:
Munged dial:
Logon time:           0
Logoff time:          never
Kickoff time:         never
Password last set:    Sun, 02 Feb 2014 03:31:44 UTC
Password can change:  Sun, 02 Feb 2014 03:31:44 UTC
Password must change: never
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
root@raspberrypi:/home#
```

图 5-17　把用户的数据输送到 Samba 上

STEP6：创建共享文件夹

共享文件夹放在 SD 卡或者外接式硬盘中，只要确定好共享路径即可，例如创建一个名为 allusers 的文件夹，提供给每个人都可以去访问，路径可以自行调整，如果使用 USB 硬盘，把/home/allusers 改为/media/usbhdd 的外接硬盘路径，如图 5-18 所示。

```
$ sudo mkdir /home/allusers
$ sudo chown root:sambashare /home/allusers/
$ sudo chmod 770 /home/allusers/
$ sudo chmod g+s /home/allusers/
```

```
root@raspberrypi:/home# sudo mkdir /home/allusers
root@raspberrypi:/home# sudo chown root:sambashare /home/allusers/
root@raspberrypi:/home# sudo chmod 770 /home/allusers/
root@raspberrypi:/home# sudo chmod g+s /home/allusers/
root@raspberrypi:/home#
```

图 5-18　创建共享文件夹

STEP7：设置 Samba

通过 nano 文本编辑器，修改 Samba 设置。

```
$ sudo nano /etc/samba/smb.conf
```

找到 Authentication,修改引导安全性,如图 5-19 所示。

请把

```
# security = user
```

改为

```
security = user
```

```
####### Authentication #######

# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server. See
# /usr/share/doc/samba-doc/htmldocs/Samba3-HOWTO/ServerType.html
# in the samba-doc package for details.
   security = user

# You may wish to use password encryption.  See the section on
# 'encrypt passwords' in the smb.conf(5) manpage before enabling.
   encrypt passwords = true

# If you are using encrypted passwords, Samba will need to know what
# password database type you are using.
   passdb backend = tdbsam

   obey pam restrictions = yes
```

图 5-19 修改引导安全性

再找到 Share Definitions,将用户由只读改为可读写,如图 5-20 所示。

把原来的

```
read only = yes
```

改为

```
read only = no
```

```
#======================= Share Definitions =======================

[homes]
   comment = Home Directories
   browseable = no

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
   read only = no

# File creation mask is set to 0700 for security reasons. If you want to
# create files with group=rw permissions, set next parameter to 0775.
   create mask = 0700
```

图 5-20 修改用户由只读为可读写

在文件的最后加上如图 5-21 所示的设置,目的是让共享的文件路径/home/allusers,可以让每个人访问,输入后按下 Ctrl+O 组合键存储和 Ctrl+X 组合键离开 nano 文本编辑器。

```
[allusers]
  comment = Shared Folder
```

```
path = /home/allusers
read only = no
guest ok = no
browseable = yes
create mask  = 0770
directory mask  = 0770
```

```
;   preexec = /bin/mount /cdrom
;   postexec = /bin/umount /cdrom

[allusers]
  comment=Shared Folder
  path=/home/allusers
  read only=no
  guest ok=no
  browseable=yes
  create mask =0770
  directory mask =0770
```

图 5-21　在文件 smb. conf 最后加上共享的文件路径/home/allusers

STEP8：重启 Samba

输入以下命令重新引导 Samba，服务器搭建完成。

```
$ sudo /etc/init.d/samba restart
```

5. 教学影片

完整的教学视频可以看 AP_FileServer。

5.3.1　Mac 连接到树莓派文件服务器

案例 20

Mac 用户如何使用 Samba 文件服务器？在 Mac 上通过 Finder 输入或选择以下网址，如图 5-22 所示。

```
smb://raspberrypi/powenko
```

输入实际所创建的用户名称和密码，连接后就可以上传和下载数据，如图 5-23 所示。

5.3.2　Windows 连接到树莓派文件服务器

案例 21

Windows 用户可以在 Windows 系统上通过文件管理器或网络邻居，输入或选择\\raspberrypi\powenko。

把 powenko 改成实际所创建的用户名称。

如果 Windows 不能通过网络邻居找到服务器，也可以通过 IE 浏览器，输入树莓派的 IP 地址和用户名称，如图 5-24 所示，输入\\192.168.0.111\powenko 就可以了。

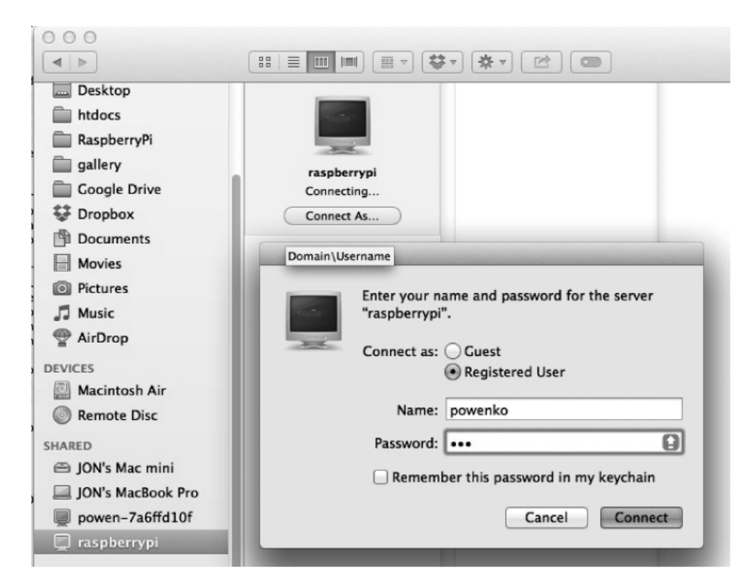


图 5-22　通过 Mac 的 finder 选择 raspberrypi 输入用户名和密码

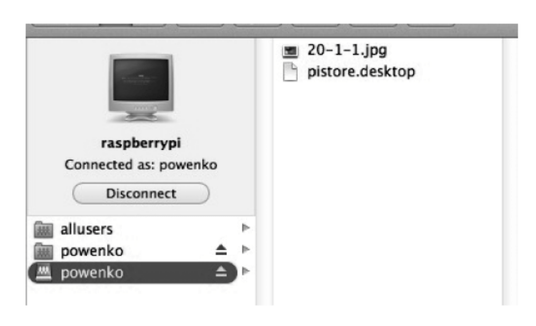


图 5-23　连接后就可以上传和下载数据



图 5-24　通过 IE 输入 IP 地址与用户名称和密码

连接后就可以上传和下载数据。可以用鼠标拖动文件夹放到桌面上,创建一个快捷方式,下回单击此图标就能使用了,如图 5-25 所示。

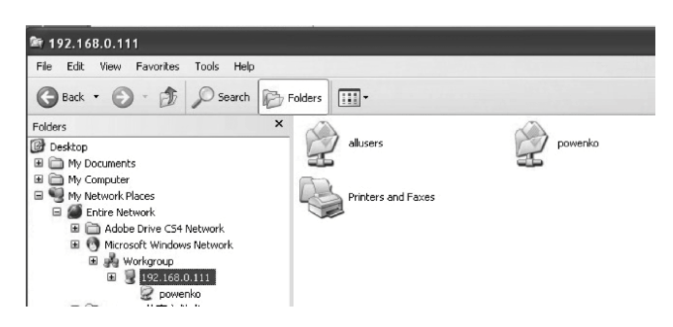


图 5-25　连接后就可以上传和下载数据

5.3.3　Linux 连接到树莓派文件服务器

案例 22

Linux 用户在 Linux 系统上通过 X-window 的文件管理器,输入或选择 smb://raspberrypi/powenko。

把 powenko 改为自己所创建的用户名称,如图 5-26 所示。

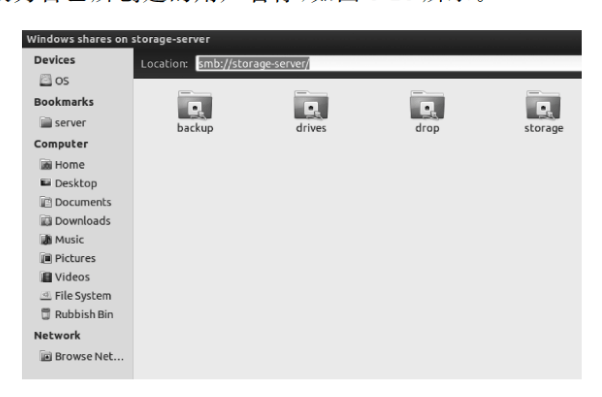


图 5-26　Linux 通过 X-window 的文件管理器连接到服务器

5.3.4　iOS 连接到树莓派文件服务器

案例 23

iOS 用户使用 iOS 的浏览器,就可以连接服务器。iOS 平台有 Samba APP,安装执行后,输入网址和密码就可以顺利执行,效果如图 5-27 所示。推荐安装 FileExplorer 这个 APP。

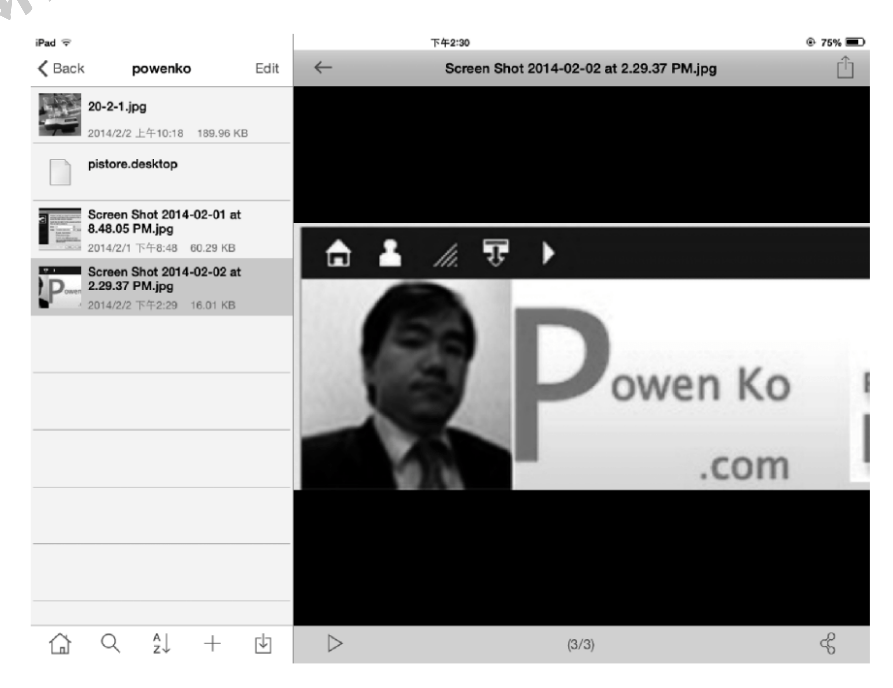


图 5-27　在 iOS 连接网络硬盘

5.3.5　Android 连接到树莓派文件服务器

案例 24

Android 用户在浏览器上输入网址，或者在 Android 平台安装 AndSMB APP 就可以连接使用了，执行后的效果如图 5-28 所示。



图 5-28　在 Android 通过 AndSMB 连接网络硬盘

通过 Android 设备到 Google Play 上查找 AndSMB,就可以找到这个免费软件,如图 5-29 所示。

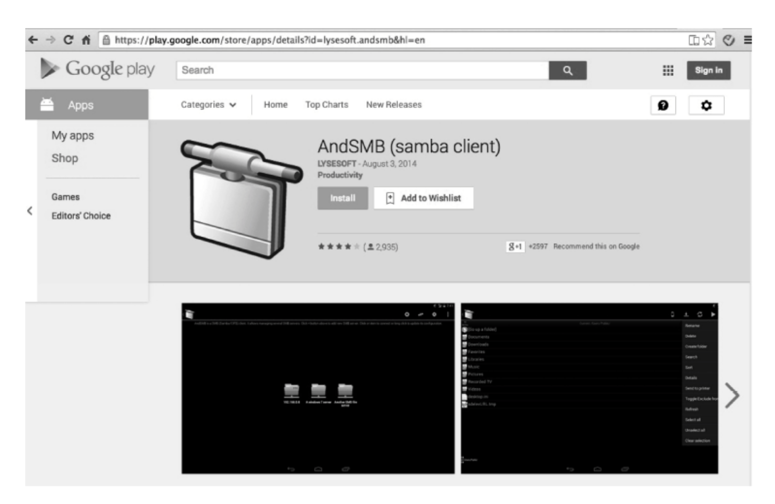


图 5-29　Google Play 上面的 AndSMB APP

5.3.6　Raspbian 连接到树莓派文件服务器

案例 25

如果有两个其他树莓派,要如何连接到树莓派文件服务器?可以在图形化窗口中 accessories\File Manger 的文件管理器中输入或选择 smb://raspberrypi/powenko。把 powenko 改成实际所创建的用户名称,如图 5-30 所示。

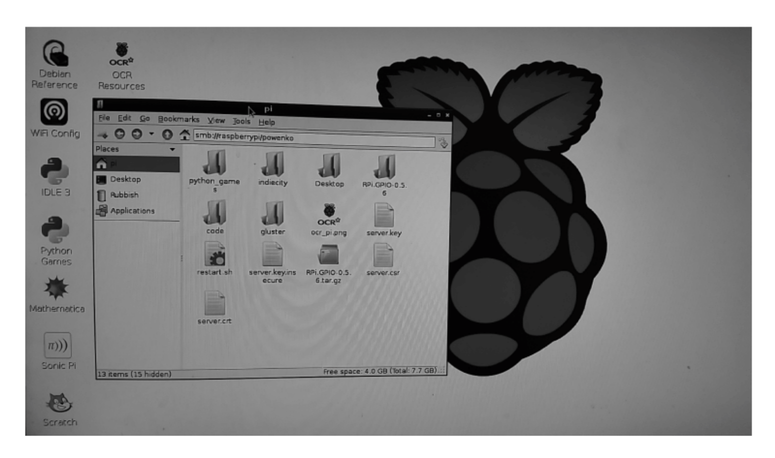


图 5-30　在 File Manger 的文件管理器输入地址

第 6 章 创建家庭云存储系统与私有云

6.1 创建云存储系统

案例 26

1. 介绍

现在很多人都会用云存储的服务,存放和备份数据,也能在不同的设备上互相访问使用,例如 Google Driver、百度云、DropBox、iCould。

但是东西放在别人的服务器上总觉得怪怪的,账号和密码一旦被破解,所有的东西都会被盗取,例如苹果的 iCould 在 2014 年 8 月就发生过一次。那么可以在家里搭建一个树莓派的云服务器吗?当然可以了!下面就会介绍怎么使用树莓派来搭建自己专用的云存储服务器。

本案例将会介绍和搭建 ownCloud,此软件可以支持多人使用和群组使用,可以设置每人、群组、分享,让云存储的服务更加便利,当前 ownCloud 支持所有的主流平台,例如网页、Android、iOS、Windows、Linux、Mac 等的操作系统平台,都有各自客户端的应用程序可以使用。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络连接设备。

会搭建以下软件:

ownCloud

3. 接线

请确认树莓派的网络功能正常。

4. 步骤

本案例是由“4.1 搭建 PHP 网站服务器”扩展而来,所以请首先搭建 PHP 网站服务器。

首先完成:

搭建 PHP 网站服务器

STEP1:下载和安装 ownCloud

通过网络下载 ownCloud 软件,如图 6-1 所示。

```
$ sudo wget http://download.owncloud.org/community/owncloud-latest.tar.bz2
```

或者是

```
$ wget http://download.owncloud.org/community/daily/owncloud-daily-master.tar.bz2
```

```
pi@raspberrypi ~/temp $ wget http://download.owncloud.org/community/daily/ownclo
ud-daily-master.tar.bz2
--2014-05-07 21:18:13--  http://download.owncloud.org/community/daily/owncloud-d
aily-master.tar.bz2
Resolving download.owncloud.org (download.owncloud.org)... 144.76.105.220, 50.30
.33.234
Connecting to download.owncloud.org (download.owncloud.org)|144.76.105.220|:80..
. connected.
HTTP request sent, awaiting response... 200 OK
Length: 32105363 (31M) [application/x-bzip]
Saving to: `owncloud-daily-master.tar.bz2'

38% [=============>                    ] 12,269,856  1.29M/s  eta 23s
```

图 6-1 下载 ownCloud 软件

如果下载源有问题,请到官方网站 http://owncloud.org/install/下载最新版本,如图 6-2 所示。

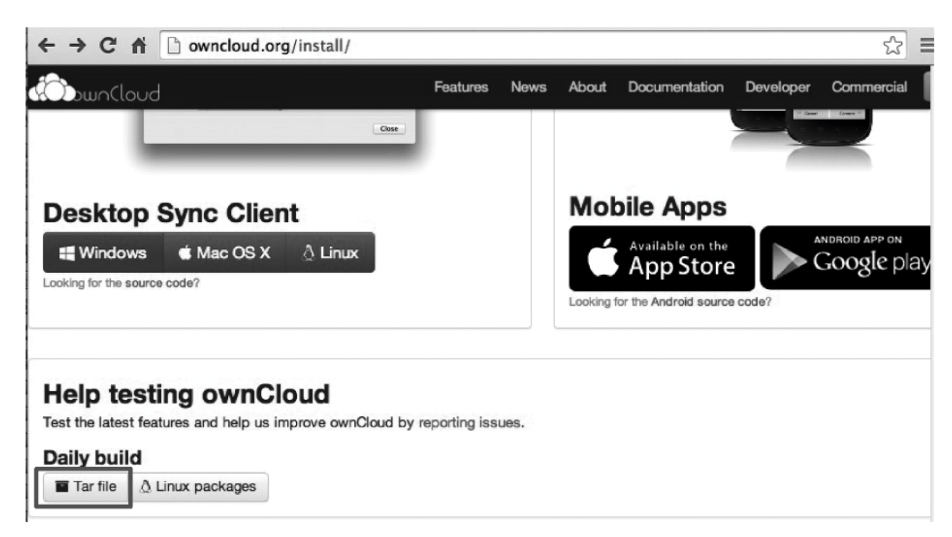


图 6-2 官方网站 ownCloud 下载软件

STEP2:解压缩

把下载的软件解压缩。

```
$ sudo tar -xjf owncloud-latest.tar.bz2
```

如图 6-3 所示,解压缩完成后,产生 ownCloud 的文件

```
pi@raspberrypi ~ $ ls
code                         owncloud-latest.tar.bz2     python_games
Desktop   owncloud          P-08:01:2014-01:06:28.jpg
```

图 6-3　解压缩后 ownCloud 的文件

STEP3:复制文件到网页路径

输入以下命令,将解压缩的文件复制到网页工作路径下。

```
$ sudo cp -r owncloud  /var/www
```

STEP4:修改网页文件的权限

让用户可以通过浏览器读取文件,需要修改文件的权限,输入以下命令来处理权限的问题。

```
$ sudo chown -R www-data:www-data /var/www/owncloud/
```

STEP5:修改 ownCloud 设置

修改 ownCloud 设置和文件大小,通过文本编辑器 nano 打开配置文件。

```
$ sudo nano /var/www/owncloud/.htaccess
```

修改参数如下:

```
php_value upload_max_filesize 1G
php_value post_max_size 1G
php_value memory_limit 1G
php_value mbstring.func_overload 0
```

通过 Ctrl+O 组合键存储,再通过 Ctrl+X 组合键离开 nano 文本编辑器,如图 6-4 所示。

```
  GNU nano 2.2.6         File: /var/www/owncloud/.htaccess              Modified

<IfModule mod_fcgid.c>
<IfModule mod_setenvif.c>
<IfModule mod_headers.c>
SetEnvIfNoCase ^Authorization$ "(.+)" XAUTHORIZATION=$1
RequestHeader set XAuthorization %{XAUTHORIZATION}e env=XAUTHORIZATION
</IfModule>
</IfModule>
</IfModule>
<IfModule mod_php5.c>
php_value upload_max_filesize 1G
php_value post_max_size 1G
php_value memory_limit 1G
php_value mbstring.func_overload 0
<IfModule env_module>
  SetEnv htaccessWorking true
</IfModule>
</IfModule>
<IfModule mod_rewrite.c>
RewriteEngine on

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 6-4　修改 ownCloud 设置

5. 结果

可以通过网页确认 ownCloud 是否设置成功,在网页浏览器中输入树莓派的 IP 地址/ownCloud,第一次进入设置好管理器的账号和密码,然后单击 Finish setup 按钮就可以进入文件管理器的主界面,如图 6-5 和图 6-6 所示。

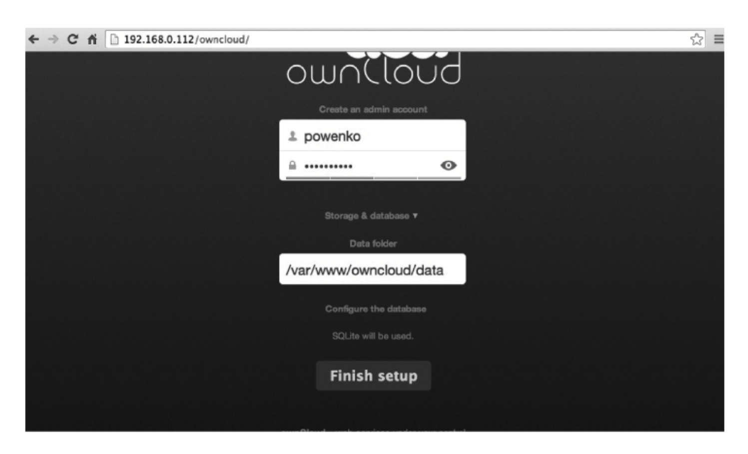


图 6-5 ownCloud 登录界面

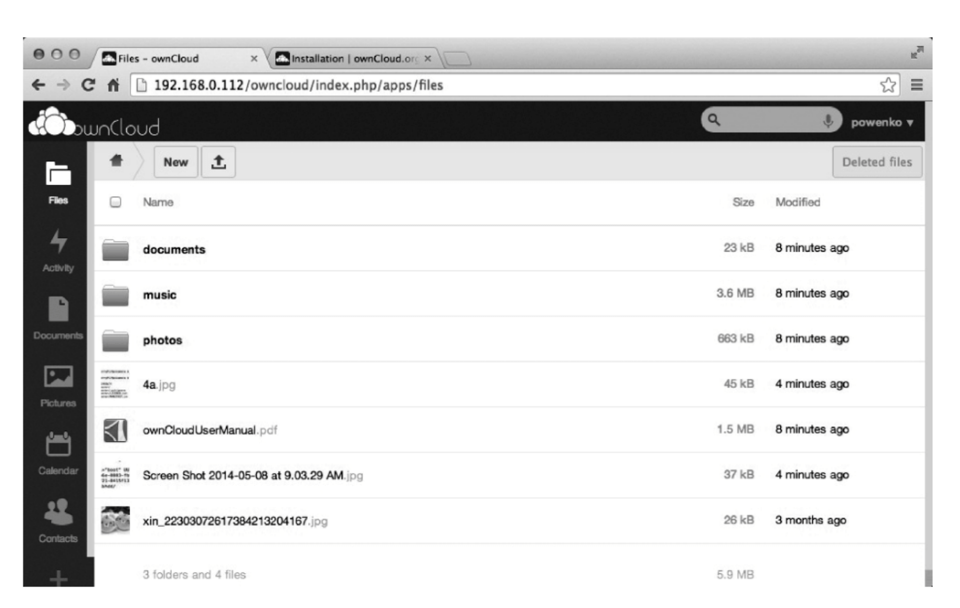


图 6-6 ownCloud 主界面

6. 教学影片

完整的教学视频可以看 raspberryPi_Owncloud_setup_t。

6.2 云数据存放在外接硬盘

案例 27

1. 介绍

当使用 ownCloud 云存储一段时间后，会觉得 ownCloud 使用的 SD 卡的空间大小，当发生 SD 卡空间不足的问题时，可以把相关数据存储在外接硬盘上吗？如果要给 ownCloud 添加硬盘，按照以下步骤执行设置。首先准备好外接硬盘并设置好。

2. 设备

需要有以下硬件

(1) 树莓派开发板；

(2) 网络连接设备；

(3) 一个 USB 外接硬盘。

需要有以下软件：

ownCloud。

同样推荐带电源的和 USB Hub，避免系统找不到硬盘的情况 USB 外接硬盘。

3. 接线

把外接硬盘接到 USB 插头上，并确认树莓派的网络运行正常。

4. 步骤

本案例是由“5.1 添加 USB 外接硬盘”、“4.1 搭建 PHP 网站服务器”、“6.1 创建云存储系统”扩展而来，所以请首先搭建 PHP 网站服务器。

需完成：

(1) 5.1 添加 USB 外接硬盘；

(2) 5.2 格式化硬盘；

(3) 4.1 搭建 PHP 网站服务器；

(4) 6.1 创建云存储系统。

STEP1：确认树莓派识别硬盘

输入以下命令确认树莓派已经识别该硬盘，可以通过硬盘名称和硬盘的大小来确认。

```
$ sudo fdisk -l
```

STEP2：添加 USB 硬盘

添加 USB 外接硬盘，注意 USB 外接硬盘文件格式需要为 FAT32，否则命令会有错误消息。

完整的设置硬盘步骤，请参考本书：

(1) 5.1 添加 USB 外接硬盘；

(2) 5.2 格式化硬盘。

STEP3：指定 ownCloud 使用外接硬盘

最好的设置方法是第一次登录 ownCloud 时，在 Data folder 指定外接硬盘的工作路径，就可以使用外接硬盘来存储数据。按照实际情况把 ownCloud 的 Data Folder 指向硬盘地址，例如/media/usbhdd/，如图 6-7 所示。

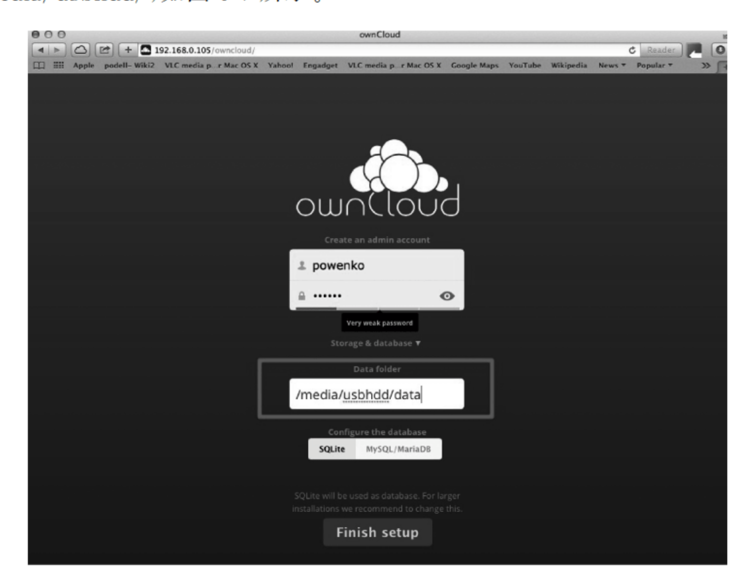


图 6-7　指定数据存放位置

输入以下命令设置硬盘路径和权限：

```
$ sudo chown -R www-data:www-data /media/usbhdd/data
```

STEP4：迁移文件

把 ownCloud 的数据库迁移或者复制到外接硬盘上，输入以下命令把 ownCloud 中的 data 文件的数据复制到外接硬盘上。

```
$ sudo su
# cd /var/www/owncloud
# mv -rf data /media/usbhdd/data
```

STEP5：设置该路径的用户权限

设置该路径的用户权限，这样才能通过网页访问文件。

```
# chown -R www-data:www-data /media/usbhdd/data
```

STEP6：修改 ownCloud 设置

修改 ownCloud 的配置文件，指定存储空间到外接硬盘上，通过文本编辑器修改 owncloud/config/config. php 这个配置文件。

```
# nano /var/www/owncloud/config/config.php
```

打开之后修改其中的工作路径并进行存储，如图 6-8 所示。

```
GNU nano 2.2.6            File: config.php                    Modified

<?php
$CONFIG = array (
  'instanceid' => 'occd0143b997',
  'passwordsalt' => '9bcd9db0c20bd3926b86074cee3f42',
  'trusted_domains' =>
  array (
    0 => '192.168.0.105',
  ),
  'datadirectory' => '/media/usbhdd/data',
  'dbtype' => 'sqlite3',
  'version' => '7.0.1.1',
  'installed' => true,
  'mail_smtpmode' => 'php',
  'mail_smtpname' => 'root',
  'mail_smtppassword' => '123456',
);

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 6-8　修改文件路径

（文件/var/www/owncloud/config/config. php）

```
<?php
$ CONFIG = array (
  'instanceid' => 'occd0143b997',
  'passwordsalt' => '9bcd9db0c20bd3926b86074cee3f42',
  'trusted_domains' =>
  array (
    0 => '192.168.0.105',
  ),
  'datadirectory' => '/media/usbhdd/data',
  'dbtype' => 'sqlite3',
  'version' => '7.0.1.1',
  'installed' => true,
  'mail_smtpmode' => 'php',
  'mail_smtpname' => 'root',
  'mail_smtppassword' => '123456',
);
```

5. 结果

设置完毕就可以通过浏览器进行更新，再一次读取 ownCloud 就会发现已经重新把文件路径指到新的外接硬盘上。

怎么知道有没有设置成功呢？可以尝试在网页上上传几个文件，然后到外接硬盘上看看这些文件是否存储在上面。可以通过以下命令看到，如图 6-9 所示。

```
# cd /media/usbhdd/data/powenko/files
# ls
```

```
root@raspberrypi:/media/usbhdd/data/powenko/files# ls
14.jpg                photos
4a.jpg                rasp.jpeg
documents             Screen Shot 2014-05-08 at 9.03.29 AM.jpg
music                 xin_18303072617382811298065.jpg
ownCloudUserManual.pdf  xin_22303072617384213204167.jpg
```

图 6-9　外接硬盘内的文件

6. 教学影片

完整的教学视频可以看 raspberryPi_Owncloud_changeHDD_t。

6.3　各操作系统使用树莓派云客户端软件

6.3.1　网页版私有云 ownCloud

案例 28

设置完毕后，可以在不同的平台上通过浏览器，例如 IE、Safari、FireFox、Chrome 等，连接该树莓派私有云的 IP 地址/owncloud，进入私有云 ownCloud 的环境中，如图 6-10 所示。

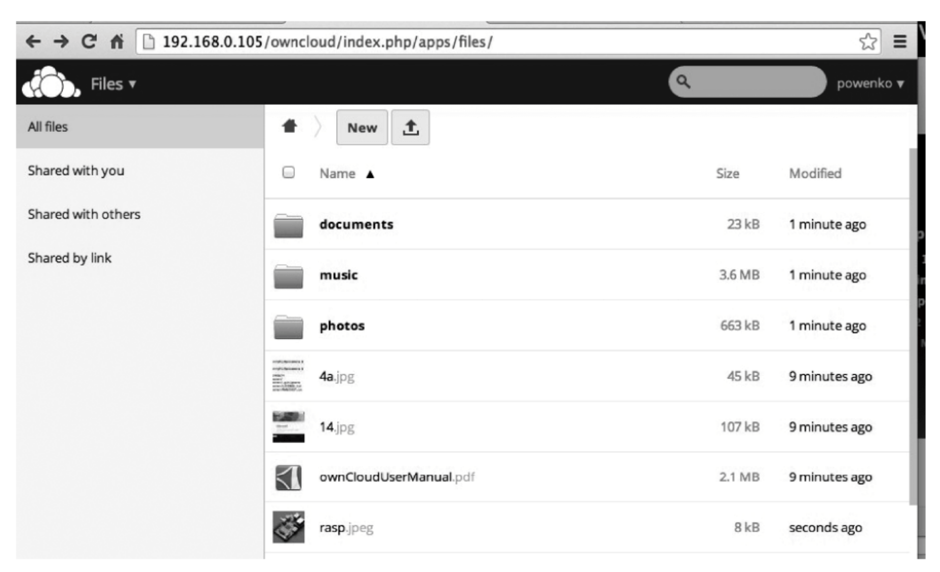


图 6-10　网页版私有云

6.3.2 Android APP 私有云软件

案例 29

ownCloud 有提供 iOS 和 Android 的 APP,可提供智能手机和平板电脑的安装与使用,Android 版本的 ownCloud APP 当前售价为 0.99 美金,可以在 Google Play 下载,只要搜索关键字 ownCloud 就能找到。下载网址为 https://play.google.com/store/apps/details?id=com.owncloud.android,如图 6-11 所示。

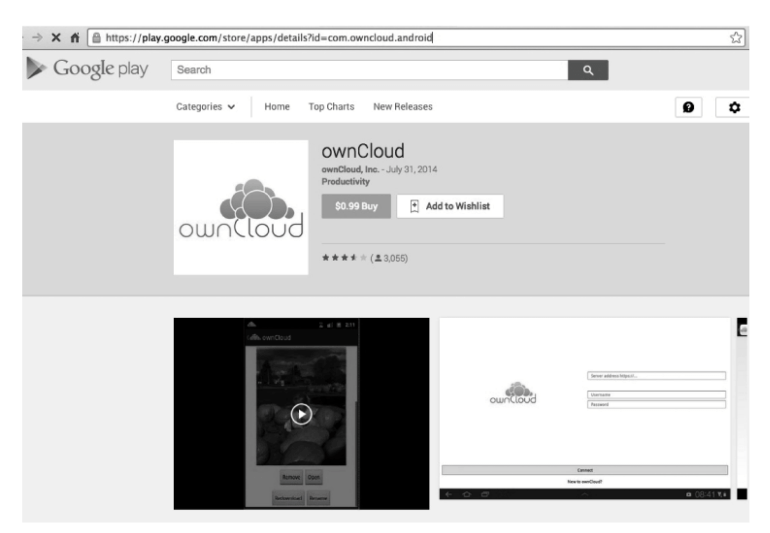


图 6-11 Google Play 上的 ownCloud APP 当前售价为 0.99 美金

下载安装后,只要设置 ownCloud 的网址,以及账号密码,就能够把手机上的数据备份到树莓派私人云上,如图 6-12 所示。

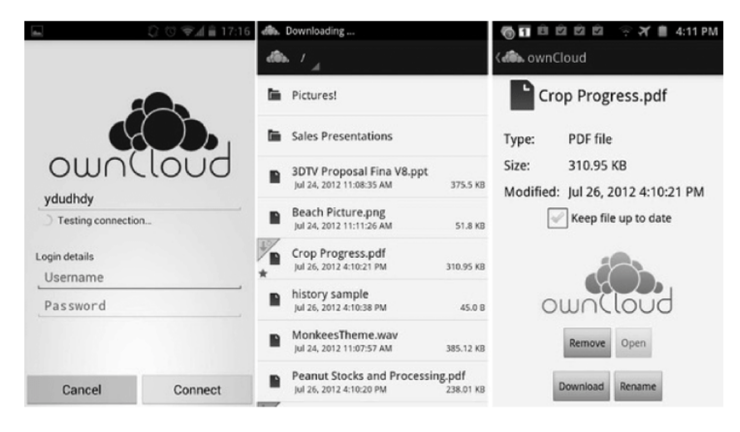


图 6-12 Android 手机的 ownCloud APP

6.3.3　iOS APP 私有云软件

案例 30

iOS 版本的 ownCloud APP 当前售价为 0.99 美金,可以在 Apple Store 下载,只要搜索关键字 ownCloud 就能找到,如图 6-13 所示。

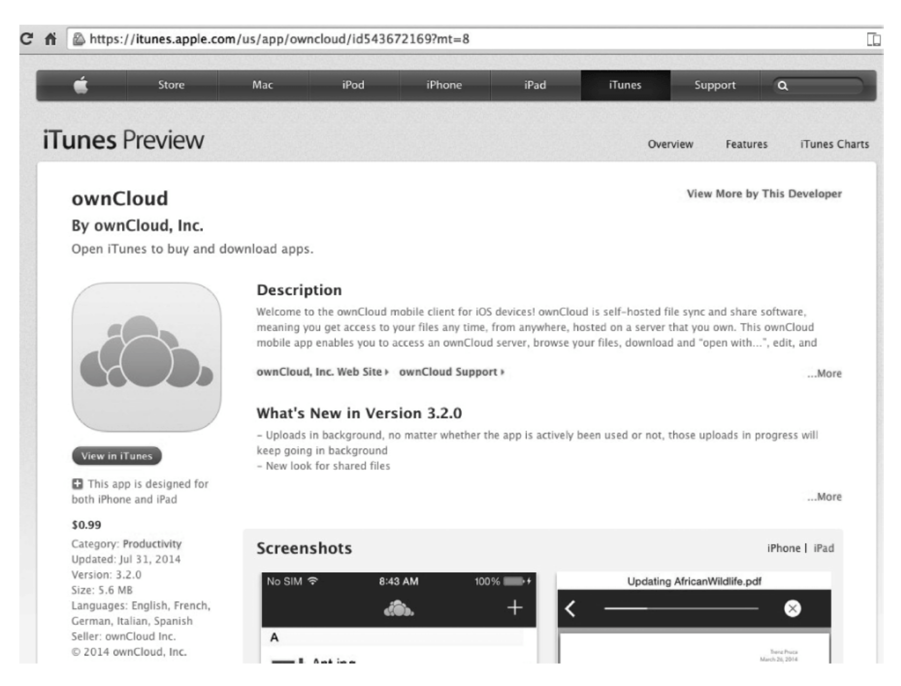


图 6-13　Apple Store 上的 ownCloud APP 当前售价为 0.99 美金

下载安装后,只要设置 ownCloud 的网址,以及账号密码,就能够把 iPhone 手机和 iPad 平板上的数据备份到树莓派私人云上,如图 6-14 所示。



图 6-14　iOS 版本的 ownCloud APP

6.3.4　Windows、Linux、Mac 版本的私有云客户端软件

案例 31

1. 介绍

Windows、Linux、Mac、iOS、Android 平台上的客户端应用程序都可以在 http://owncloud.org/install/下载,如图 6-15 所示。

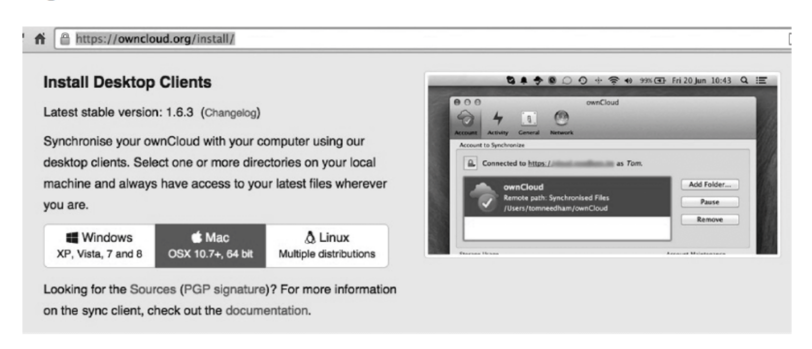


图 6-15　Windows、Linux、Mac 平台应用程序下载的地方

2. 步骤

STEP1：下载安装软件

通过网页连接到 http://owncloud.org/install/,下载对应操作系统的软件,解压缩后,进行安装。

STEP2：设置 ownCloud

Mac、Windows、Linux 操作系统,第一次执行 ownCloud 软件时,都会询问用户的账号与密码,如图 6-16 所示。



图 6-16　设置 ownCloud

STEP3：设置 ownCloud 的网络地址

设置树莓派 ownCloud 的网络地址，并且指定在客户端所存储的路径，如图 6-17 所示。

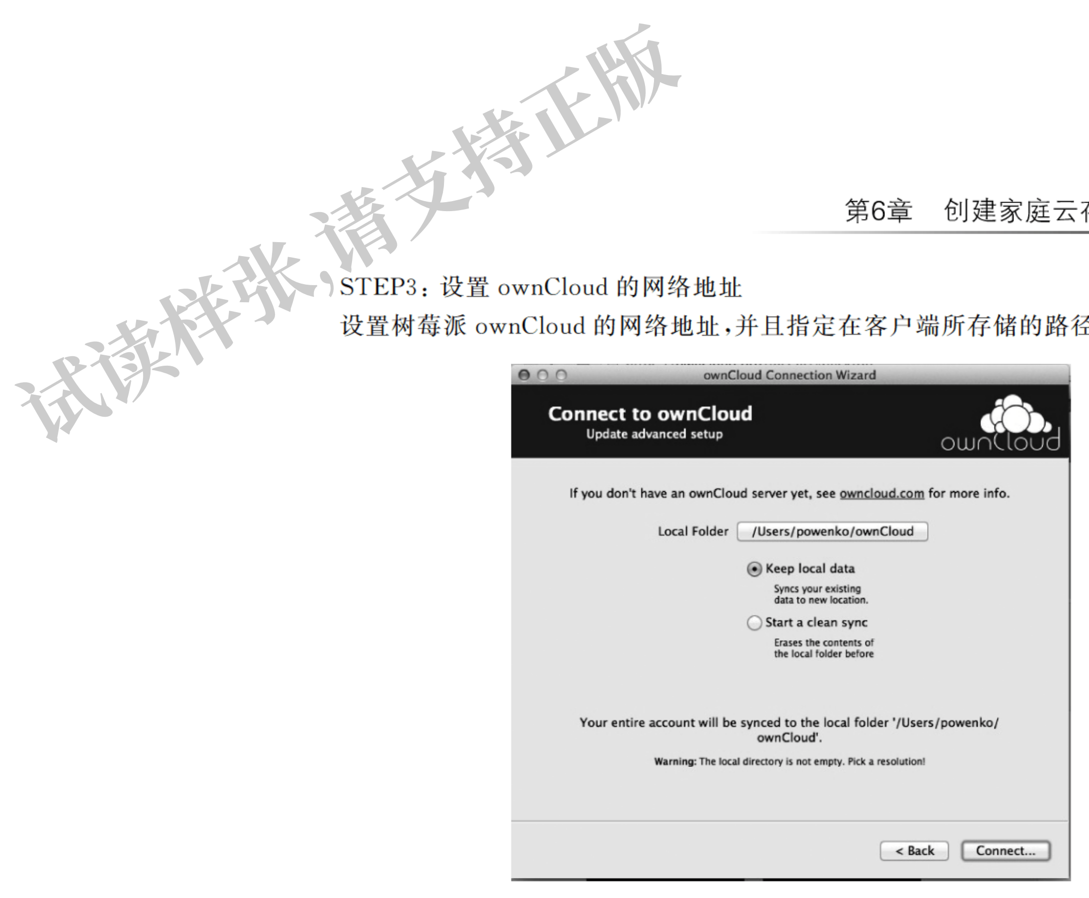


图 6-17　设置 ownCloud 的网络地址

3. 结果

设置完成，以后就可以把文件直接放在本机客户端的路径中，ownCloud 会自动同步，如图 6-18 所示。



图 6-18　自动同步

第 7 章 树莓派机顶盒

把树莓派调整为电视机的多媒体播放机，这样就可以让家人在客厅的大电视机上面，通过树莓派的多媒体播放机，一起观看网络电影、手机上的照片等。

一般，搭建电视机专用的树莓派播放系统有 4 种选择，分别是：

(1) Raspbmc；

(2) OpenELEC；

(3) XBian；

(4) RasPlex (Plex)。

每一种操作系统都有各自的首选项，而且 Raspbmc、OpenELEC、XBian 都是由 XBMC 系统发展而来，RasPlex 是由 Plex 调整后的版本，其目的都是播放多媒体文件，功能都相差不多。本书以 OpenELEC 为例，RasPlex 执行界面如图 7-1 所示。原因是它的安装与设置是最方便的，对读者来说最容易上手。

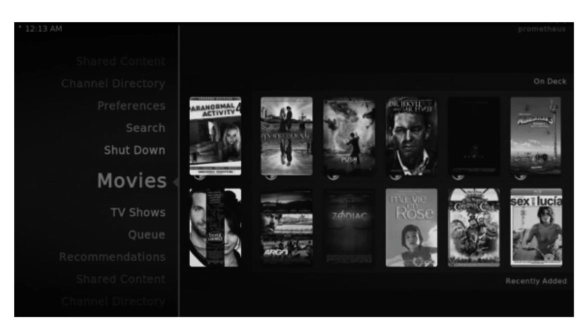


图 7-1　RasPlex 执行的界面

7.1　安装 img 镜像文件

案例 32

1. 介绍

本书在开始的时候，已经介绍过如何使用 NOOBS 安装树莓派操作系统，本章会介绍安

装电视机专用的操作系统 OpenELEC。虽然 NOOBS 也可以安装 OpenELEC,但是想特别用另外一个方法,即通过 img 镜像文件制作开机的 SD 卡,因为大多数的树莓派操作系统都是使用 img 镜像文件,所以通过这样的方法来安装操作系统是一定要学习的。

2. 步骤

STEP1:下载 OpenELEC 镜像文件

通过浏览器连接到 http://www.raspberrypi.org/downloads/,如图 7-2 所示,下载最新版本的 OpenELEC 的 ZIP 文件,文件名为 OpenELEC-RPi.arm-4.0.5.img.gz,解压缩后文件名为 OpenELEC-RPi.arm-4.0.5.img。

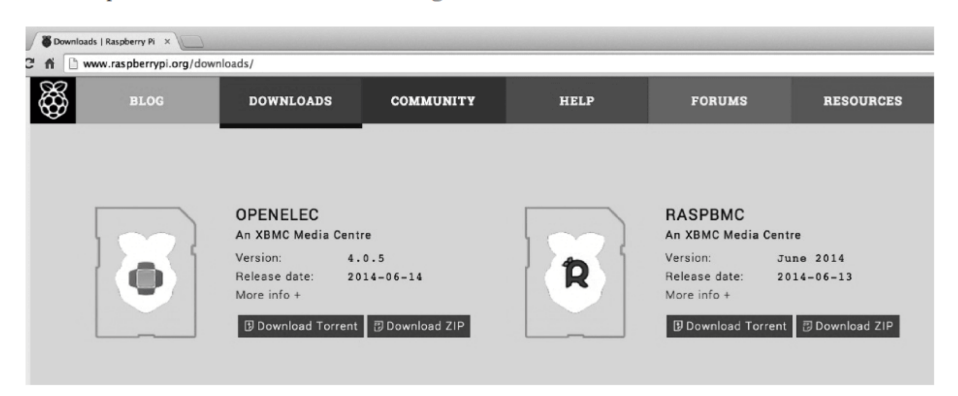


图 7-2 下载 OpenELEC

STEP2:下载 Win32DiskImager 工具

为了把镜像文件 img 刻录到 SD 卡上,按照下面的操作下载 Win32DiskImager 软件,如图 7-3 所示。首先输入 https://launchpad.net/win32-image-writer,单击 External downloads 就可以下载 image-writer 工具了。

选中下载文件单击鼠标右键,解压缩这个 ZIP 文件,就可以看到 Win32DiskImager,如图 7-4 所示。

STEP3:执行 Win32DiskImager 工具

按照图 7-5 所示,执行以下操作:

(1) 把 4GB 以上容量的 SD 卡插入 Windows 系统的计算机。

(2) Windows Vista、Windows 7 和 Windows 8 操作系统的用户,推荐单击鼠标右键,选中 Run as administrator 执行程序 Win32DiskImager.exe,如果是 Windows XP 就直接打开执行。

(3) 指定 img 镜像文件和 SD 卡的位置。

(4) 单击 Write 按钮。

(5) 如果 SD 卡内有数据,会弹出是否要清除数据的对话窗,单击 Yes 按钮。

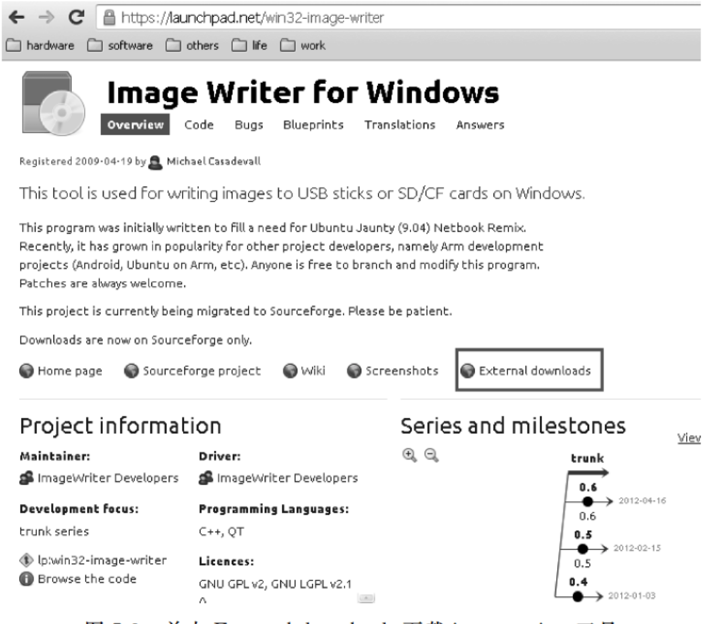


图 7-3 单击 External downloads 下载 image-writer 工具

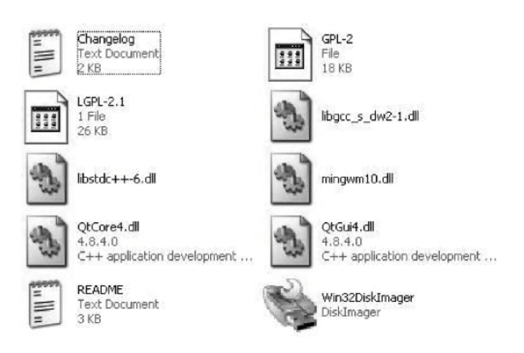


图 7-4 Win32DiskImager 文件夹

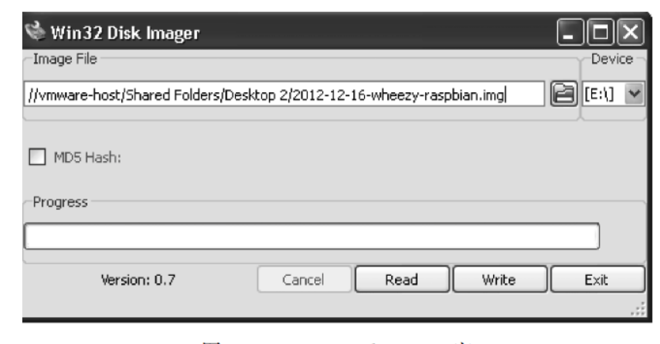


图 7-5 Win32DiskImager 窗

STEP4：执行完成

接下来会看到滚动条由0%到100%的处理过程,完成后出现Complete窗口,单击OK按钮就可以了,如图7-6所示。

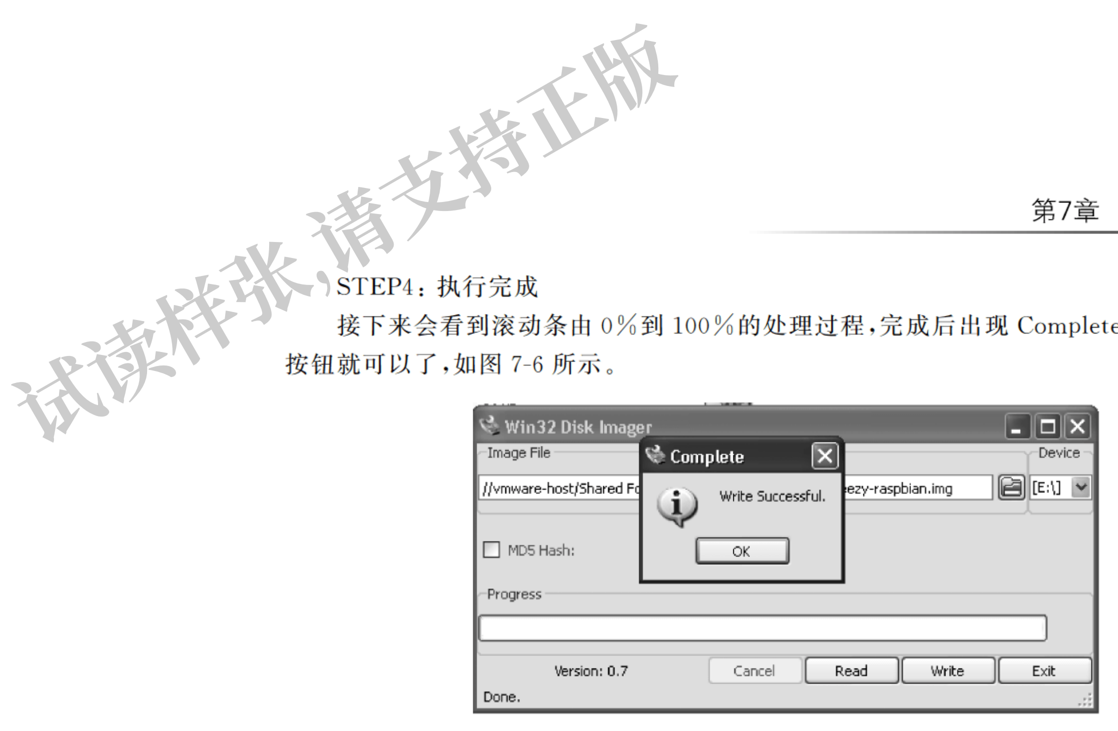


图7-6　完成后出现Write Successful窗口

STEP5：确认

查看SD卡的数据是否存在,如图7-7所示,确认后就可以把SD卡取出,插在树莓派上。

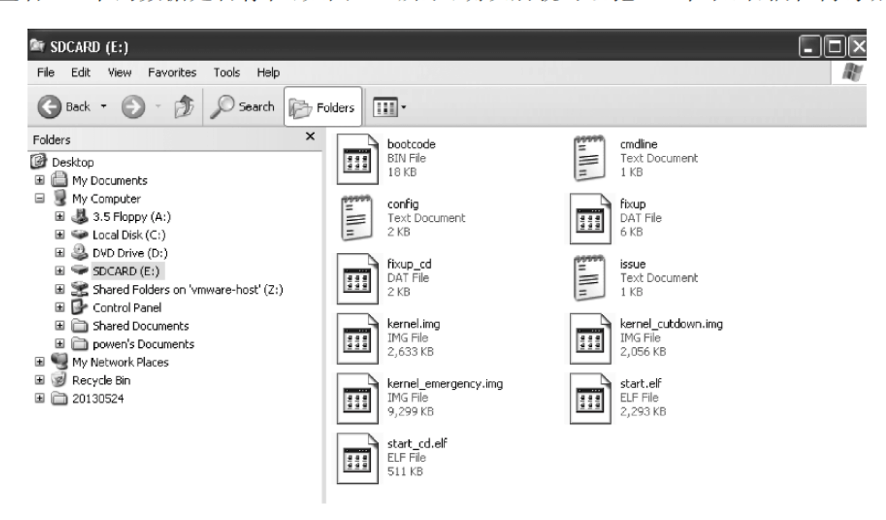


图7-7　通过文件管理器查看SD卡的数据是否存在

7.2　树莓派机顶盒开机和设置

案例33

1. 目的

当第一次把安装过OpenELEC的img镜像文件刻录成SD卡,并把该SD卡放入树莓

派开机后,需要做一些设置。

首先 OpenELEC 的 img 镜像文件管理器权限的账号为 root,密码是 OpenELEC。

2. 步骤

STEP1:设置语言

第一次进入 OpenELEC 时,如图 7-8 所示进行语言设置,虽然有简体中文 Chinese (Simple)和繁体中文 Chinese (Traditional)的选项,但还是请选择英文 English,详细的中文设置,因为步骤有些繁琐,后面章节会有详细介绍,选择后,单击 Next 按钮,进入下一个设置。



图 7-8 设置语言

注意 请先不要设置中文,因为缺少字体库,会使所有主菜单会变为空白。

STEP2:设置名称

为了在网络上方便查找这台树莓派机顶盒,可以设置它在的网络上的名称,单击 Next 按钮,到下一步操作,如图 7-9 所示。

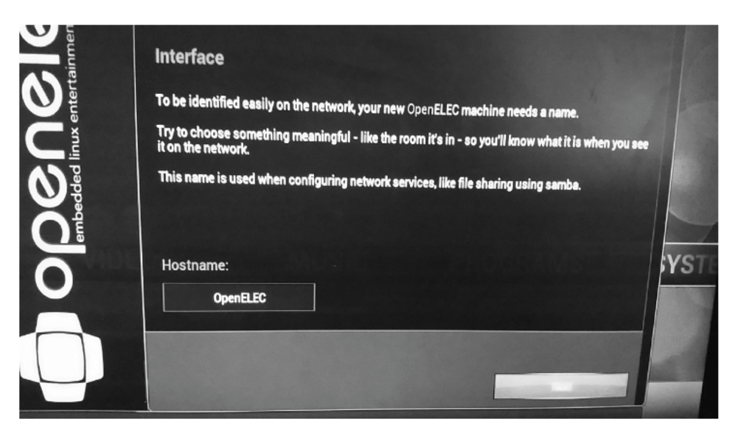


图 7-9 设置名称

STEP3：设置网络连接

如果已经把网线连接到树莓派上，这里会看到动态 IP 地址，单击该选项作为默认的连接设置，如果需要设置无线网络连接，也在此界面设置密码。完毕后，单击 Next 按钮，进入下一个设置，如图 7-10 所示。

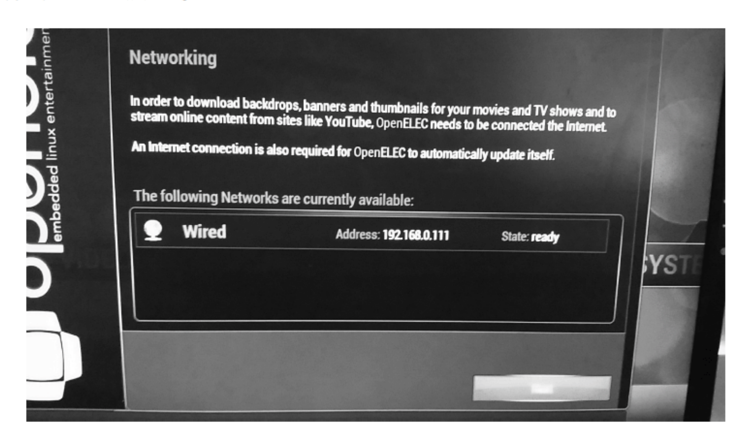


图 7-10　指定连接的设备

STEP4：设置远程连接的方式

这里设置通讯协议的方法，如图 7-11 所示，可以在计算机和手机上连接到这台树莓派，方便以后传递文件和数据，这里 OpenELEC 有两种方法：

(1) 一种是 SSH，详细介绍见作者另一本著作《Raspberry Pi 最佳入门与实战应用》中的“3.4 使用 SSH 远程控制树莓派”。

(2) 另一种是 Samba，详细介绍见本书的“第 5 章网络附加存储 NAS 系统与 USB 外接硬盘文件服务器”。

这两种方法都要设置，完成后，单击 Next 按钮，到下一步操作。



图 7-11　设置远程连接的方式

STEP5：完成

接下来会出现完成设置的窗口，同样单击 Next 按钮，就会顺利进入主界面。

3. 结果

整个设置完毕后，就可以进入 OpenELEC 的主界面，如图 7-12 所示，也就是 XBMC 的操作界面，接下来就可以通过键盘上的上、下、左、右、Enter 和 Esc 键来控制，鼠标也可以控制和点选。



图 7-12 主界面

4. 教学影片

完整的教学视频可以看 Raspberrypi_XBMC_init_T。

7.3 设置音频输出

案例 34

1. 目的

默认的音频输出是通过 HDMI，如果想设置用树莓派上的 Audio Jack(耳机麦克共用接头)输出，需要通过以下步骤设置。

2. 步骤

STEP1：进入设置

请通过键盘或鼠标，选择 OpenELEC 主界面菜单的 System→Settings，如图 7-13 所示。

STEP2：设置

进行以下的设置修改，如图 7-14 所示。

(1) 在设置界面，选择 System。

(2) 在 Audio output 声音输出中，选择 Audio output device，把原来的 HDMI 调整为树莓派上的设备输出，Analogue 为 Audio Jack 耳机麦克输出，HDMI 是通过电视机 HDMI 输



图 7-13　进入设置



图 7-14　设置音效

出声音。其他的选项是：

① Number of channels：设置为几声道输出。

② Keep audio device alive：设置几分钟持续声音输出。

③ Play GUI sounds：设置界面的声音。

④ Reset above settings to default：撤销设置。

因为 XBMC 对外接硬盘和 U 盘的支持非常好，只要接上去，系统就会自动挂载设备直接使用，所以可以把一些影片或音乐复制到 USB 存储设备上，并接到树莓派测试查看是否设置成功。

3. 教学影片

完整的教学视频可以看 Raspberrypi_XBMC_setup_audio_T。

7.4　设置中文

案例 35

1. 目的

树莓派本身需要用中文，但需要指定字体库才能顺利使用，如果字体库没有支持中文，

菜单栏会出现空白。请按照本案例的介绍,设置 XMBC 支持中文显示。

2. 步骤

STEP1:进入设置

通过键盘或鼠标,选择 OpenELEC 主界面菜单栏的 System→Settings。

STEP2:进入 Appearance 外观设置

通过键盘或鼠标,选择 Appearance 进入外观设置,如图 7-15 所示。

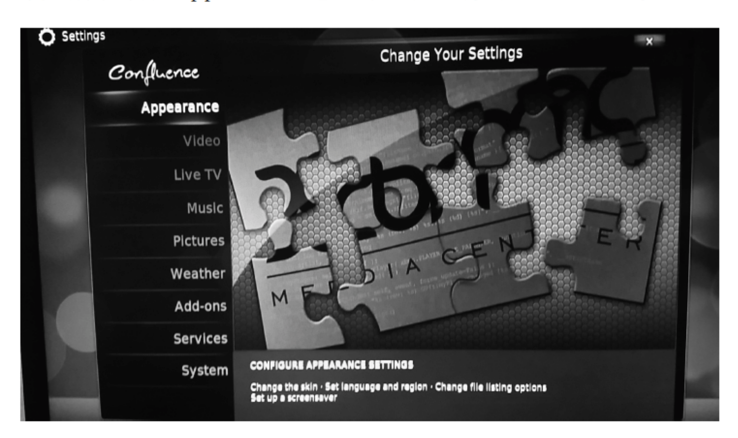


图 7-15 进入 Appearance 外观设置

STEP3:设置字体

在外观设置界面的 Skin→Fonts,选择 Arial based,如图 7-16 所示。



图 7-16 设置字体

STEP4:指定语言

在 International→Language 设置为 Chinese(Simple)或 Chinese(Traditional),看个人

的使用习惯,设置完毕等5秒钟,就会切换成功,如图7-17所示。



图7-17　设置语言

3. 结果

设置完毕后整个界面都会显示中文,如图7-18和图7-19所示。



图7-18　中文繁体 Chinese(Traditional)



图7-19　中文简体 Chinese(Simple)

4. 教学影片

完整的教学视频可以看 Raspberrypi_XBMC_setup_chinese。

7.5 更改外观

案例 36

1. 目的

XBMC 自带了很多不同样式的外观,可以供大家选择,本案例将会介绍如何设置外观和换肤。

2. 步骤

STEP1:进入设置

请通过键盘或鼠标,选择 System→Settings,进入 Appearance-Settings 界面。

STEP2:设置外观

单击 Skin→Skin,默认的外观为 Confluence,单击 Confluence 进入设置项,如图 7-20 所示。



图 7-20 设置外观

STEP3:获得更多的外观样式

单击 Get More 获得更多的外观样式,如图 7-21 所示。

STEP4:外观介绍

在如图 7-22 所示列表中,可以挑选喜欢的样式,选中后会显示该样式的介绍。

STEP5:下载

确定后就可以单击 Install 按钮进行安装,如图 7-23 所示,整个安装的用时间不定,请耐心等待。

STEP6:切换

下载完成后,就可以回到 Skin 换肤的地方进行界面切换,选中想要的样式,如图 7-24 所示,确认后系统便会自动切换新的外观。



图 7-21 获得更多的外观样式

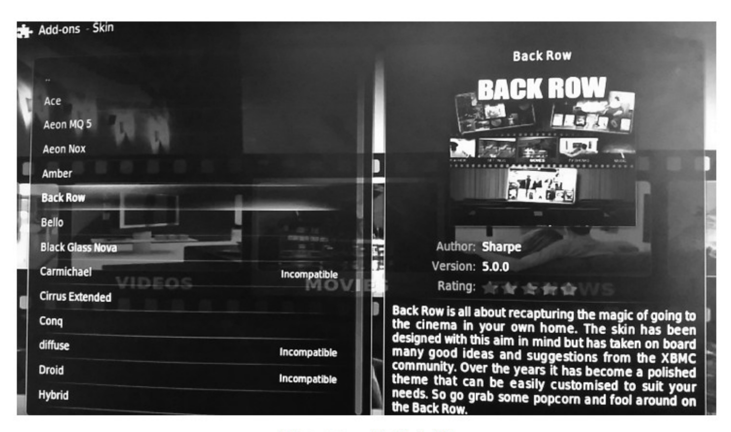


图 7-22 外观介绍



图 7-23 下载安装



图 7-24　选择样式

3. 结果

稍等之后,XBMC 的外观会按照所选的外观重新设置,如图 7-25 和图 7-26 所示。

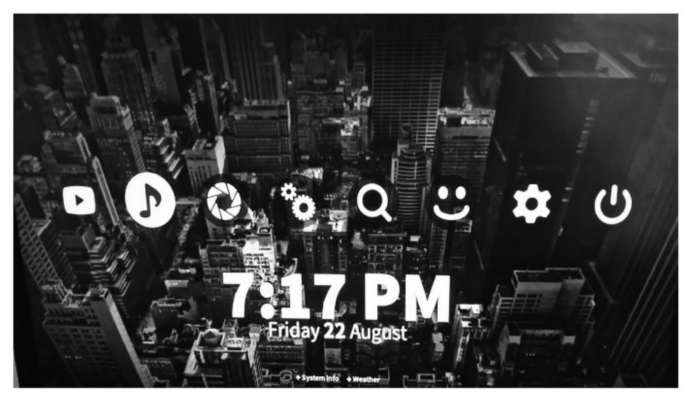


图 7-25　Maximinimalism 的外观

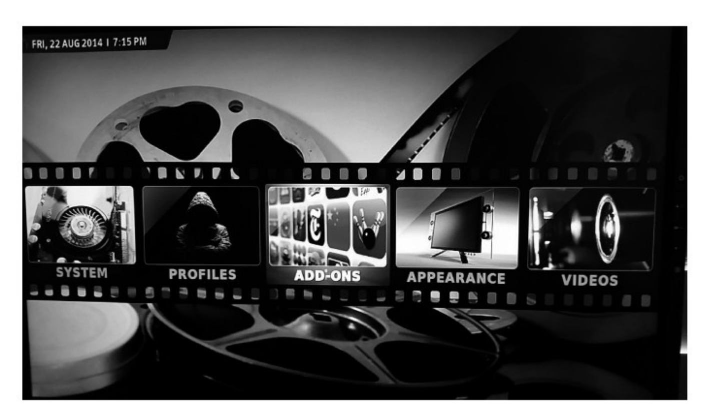


图 7-26　Back Now 的外观

4. 教学影片

完整的教学视频可以看 Raspberrypi_XBMC_ChangeSkin。

7.6　中文视频和音乐

案例 37

1. 目的

因为希望树莓派机顶盒可以支持中文的广播和电视,所以这个案例介绍如何设置中文的多媒体。

2. 步骤

STEP1:进入设置

在浏览器中输入网址 https://code.google.com/p/xbmc-addons-chinese/,到下载区 Download 中查找 Chinese Add-on Repository for XBMC eden,然后下载,如图 7-27 所示。

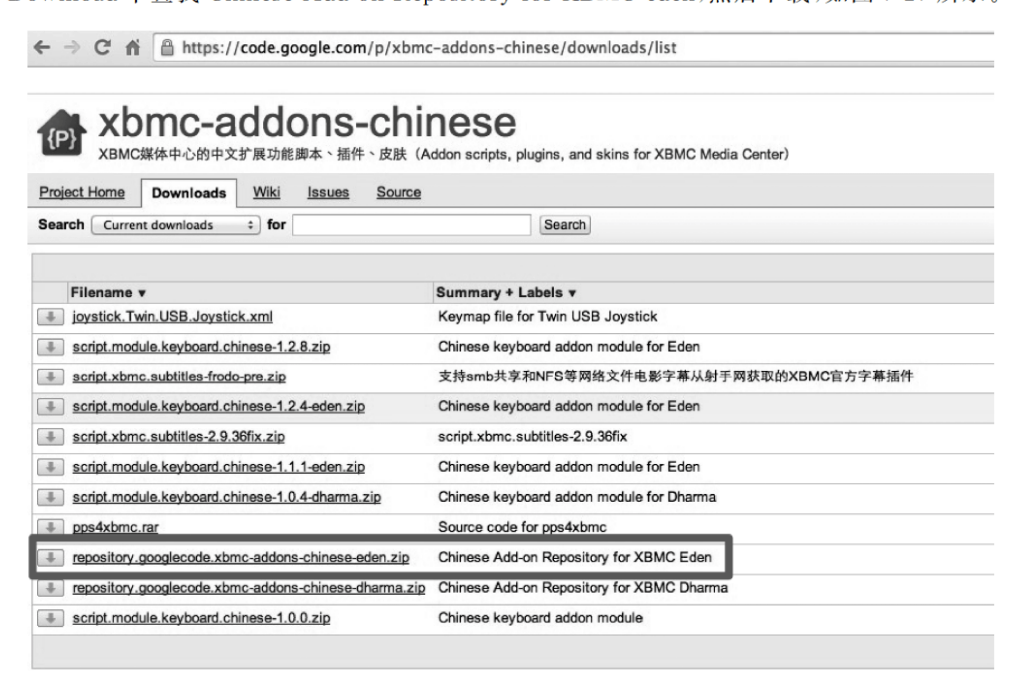


图 7-27　xbmc-addons 中文下载

STEP2:把 zip 文件复制到 USB 外接硬盘或 U 盘

把刚刚下载的 zip 文件复制到 USB 外接硬盘或 U 盘中,并把该 USB 外接硬盘或 U 盘连接到树莓派 USB 接口上。

STEP3：安装

通过键盘或鼠标，在 OpenELEC 中选择 System→Settings，然后选中 Add-ons 如图 7-28 所示。

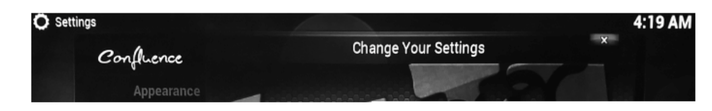




图 7-30　指定到实际的 USB 存储设备

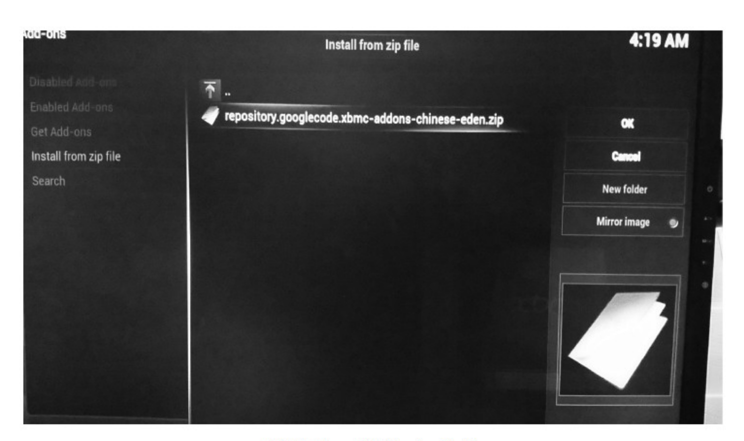


图 7-31　指定 zip 文件



图 7-32　选择 Videos 的 Add-ons 添加内容

选择 get more 就会看到等待安装的影片内容,如图 7-33 所示,往下就会看到中文的内容,选中后就可以安装网站和中文媒体。



图 7-33　中文的媒体

3. 结果

等安装完毕,再进入 Videos 就可以看到刚刚选择的网站内容,如图 7-34 所示,例如土豆网,并且可以按照分类找到想看的网络影片。



图 7-34　视频列表

播出效果非常好,原因是树莓派的多媒体是通过硬件解码,所以可以播放出高品质的影片,如图 7-35 所示。

4. 教学影片

完整的教学视频可以看 Raspberrypi_XBMC_setup_chinese_addone。

注意　同样,在 Music→Add-ons 的 get mores 中,如图 7-36 所示,也能够听到中文的音乐和联机广播。



图 7-35 影片播放情况

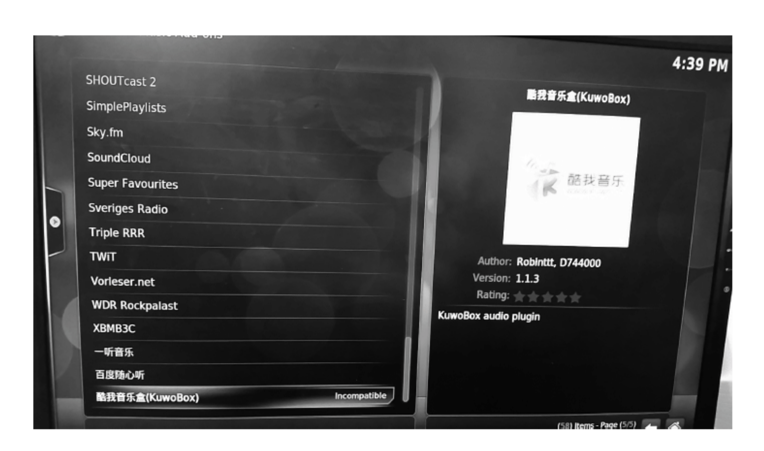


图 7-36 中文音乐列表

7.7 推荐树莓派机顶盒的外挂 Add-ons 程序

7.7.1 官方外挂

案例 38

1. 目的

在 OpenELEC 中有很多很棒的 Add-ons 外挂，再来推荐一些好的 Add-ons 外挂。并且在网络上也有很多非官方提供的 Add-ons 外挂，可以通过 System→Add-ons→install from zip file 先下载后安装。

2. Music 音乐类

Pandora：潘多拉联机音乐电台，内容丰富且合法，但是限美国使用，如图 7-37 所示。

TuneIn Radio：包括很多电台的网站，其中还有华人圈的电台。

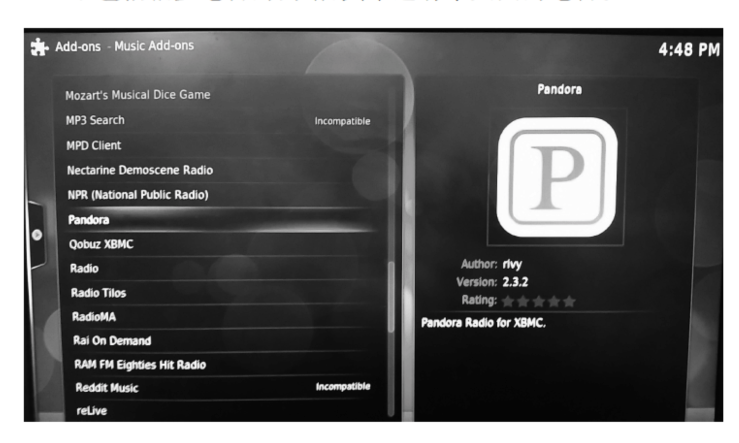


图 7-37　潘多拉联机音乐电台

3. Video 影片类

Ted talk：相信很多人已经在网络上看过这个专题演讲的网站，XBMC 中也有这个外挂，安装之后就可以在电视机上配合树莓派机顶盒观看演讲。

Food Network：介绍美食的电视节目。

Fox News：福克斯电视网，新闻频道，如图 7-38 所示。

YouTube：也有外挂程序可以在电视机上观看影片。

NetFlixbmc：联机付费点播电影。



图 7-38　福克斯电视网，新闻频道

4. System 系统类

Unofficial OpenELEC Mediacenter OS Add-ons：里面有很多各式各样非官方的外挂。

7.7.2 非官方外挂 XBMC SuperRepo

案例 39

1. 目的

如果还觉得不过瘾，可以下载非官方的外挂程序 XBMC SuperRepo，如图 7-39 所示，里面有 1300 个外挂可以使用。

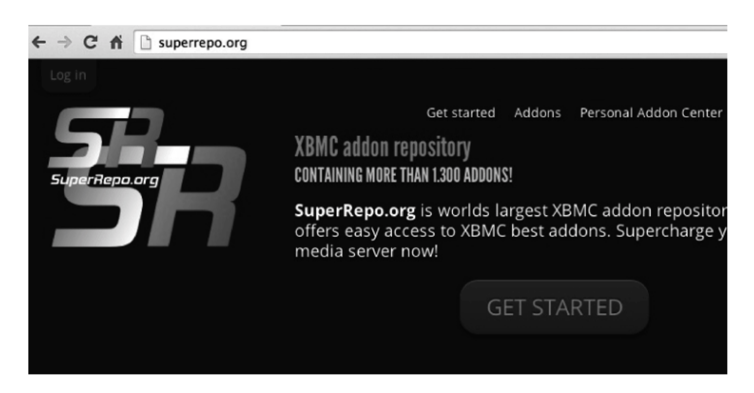


图 7-39 XBMC SuperRepo 的官方网站

2. 步骤

STEP1：安装——主菜单中的 SYSTEM→File manager

在 XBMC 主菜单中单击 SYSTEM→File manager 选项，如图 7-40 所示。

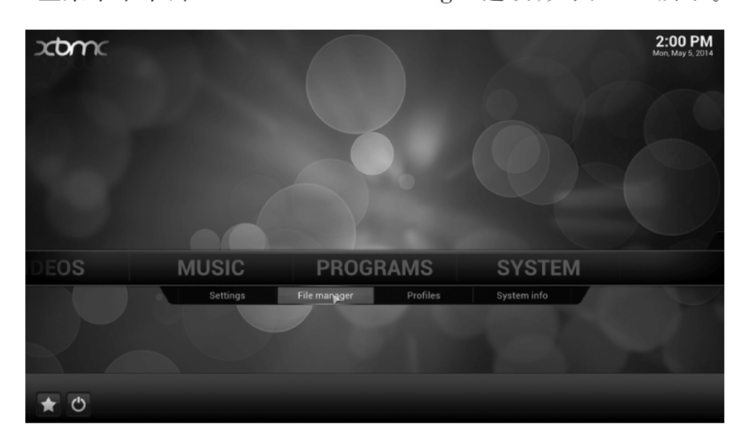


图 7-40 SYSTEM→File manager

STEP2：安装——单击 Add Source 中的 None

选择界面左侧导航栏的 Add Source，单击 None 选项，如图 7-41 所示。



图 7-41　单击 Add Source 中的 None

STEP3：安装——输入 http://srp. nu 和 SuperRepo

输入 http://srp. nu，然后单击 Done。在下面的名称框中输入 SuperRepo，之后单击 OK 按钮完成，如图 7-42 所示。



图 7-42　输入网址 http://srp. nu 和名称 SuperRepo

STEP4：安装——进入 System→Addons

回到主界面，再进入 System，然后选择 Add-ons，如图 7-43 所示。

STEP5：安装——进入 Install from Zip

选择 Install from Zip，就能够看到 XBMC 版本的 SuperRepo 数据库，如图 7-44 所示。

STEP6：安装——选择 ZIP

选择 gotham→All→repository. superrepo. org. gotham. all-0. x. x. zip，单击 OK 按钮，如图 7-45 所示。



图 7-43　进入 System→Add-ons



图 7-44　进入 Install from Zip 中的 SuperRepo



图 7-45　选择 gotham→All→repository. superrepo. org. gotham. all-0. x. x. zip

3. 结果

完成后,可以在 Video 和 Music 的 Add-one→Get More 中看到接近 1300 个外挂,如图 7-46 所示。



图 7-46　超多的外挂可供使用

4. 教学影片

完整的教学视频可以看 Raspberry_TV_ SuperRepo。

7.8　分享文件

案例 40

在这里介绍几种方法让个人计算机、手机、平板电脑上的多媒体影音数据,分享到 XBMC 树莓派机顶盒上来播放和观看。

首先让我们搞清楚什么是 UPnP? UPnP(Universal Plug and Play)是针对家庭的多媒体文件分享机制,它可以分成多种模式,最常见的是:

(1) UPnP Server:多媒体的服务器,负责文件的存储和来源。

(2) UPnP Player:多媒体播放器,负责文件的播放和显示。

(3) UPnP Controller:遥控器。

7.8.1　树莓派机顶盒 UPnP 分享文件

打开 UPnP Server 分享文件

依次单击 System→settings→Service→UPnP,把以下三项全部打开,如图 7-47 所示。

(1) Share video and music libraries through UPnP:通过 UPnP 分享影片和音乐。

(2) Look for remote UPnP players:查找其他的 UPnP 播放器。

(3) Allow control of XBMC via UPnP:允许通过 UPnP 的方法来控制树莓派机顶盒,



图 7-47 打开 UPnP Server 分享文件

这项功能的用意是把树莓派里面的多媒体文件,分享给其他设备。

设置完毕,重新开机。

7.8.2 在 Android 操作系统播放树莓派上的影音

案例 41

1. 目的

播放树莓派机顶盒上的多媒体影音数据。

2. 步骤

STEP1:安装 UPnP APP

在 Android 操作系统上安装 UPnP Player,推荐 Google Play 下载一套免费的 UPnP/DLNA control 的软件,叫做 BubbleUPnP UpnP/DLNA,它有免费版和付费版,可以试用免费版,如图 7-48 所示,差异是免费版只能在 playlist 播放 16 首歌曲。推荐原因是这个软件包含:

(1) UPnP Server:多媒体的服务器,负责文件的存储和来源。

(2) UPnP Player:多媒体播放器,负责文件的播放和显示。

STEP2:安装播放树莓派上的多媒体软件

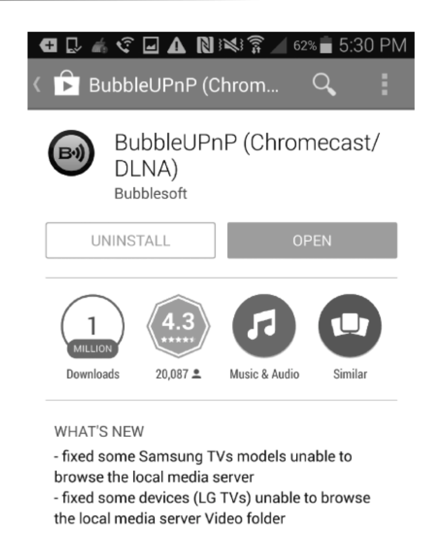
完成后打开执行 BubbleUPnP,首先单击左上角的三重条线,如图 7-49 所示,用来打开设置。

STEP3:在手机上播放树莓派机顶盒的多媒体

将配置文件调整为如图 7-50 所示。

(1) Render 显示为 Local Renderer 的意思是设置本身为播放,也就是安装此软件的手机会成为播放端。

(2) Library 数据来源设置为 XBMC 树莓派机顶盒。



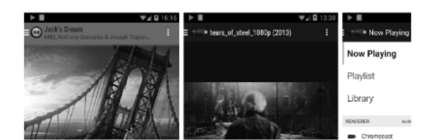


图 7-48　安装 Android 系统的 UPnP/DLNA 程序

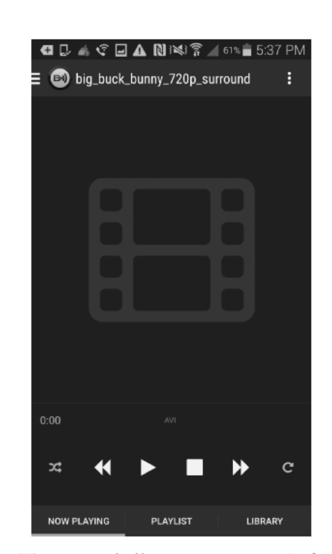


图 7-49　安装 BubbleUPnP 程序

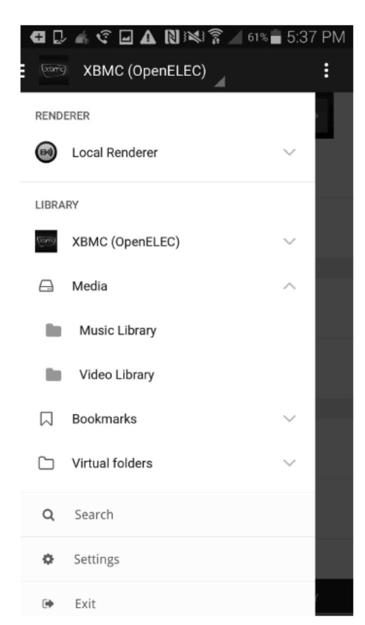


图 7-50　设置 UPnP

3. 结果

单击下方的 Media 多媒体文件，会显示出树莓派机顶盒的多媒体文件内容，单击喜欢的文件就可以在手机上播放，如图 7-51 所示。



图 7-51　播放多媒体文件

7.8.3　在树莓派上播放 Android 操作系统的影音

案例 42

1. 目的

在树莓派机顶盒播放 Android 手机的多媒体影音。

2. 步骤

继续案例 4.1 的步骤往下执行。

STEP4：在树莓派机顶盒播放手机的多媒体

把手机上的多媒体，在树莓派机顶盒播放，将 BubbleUPnP 设置调整为：

(1) Render 显示播放设置为 XBMC 树莓派机顶盒。

(2) Local Renderer 手机端。

(3) Library 数据来源设置为 Local Renderer 手机端。

3. 结果

在 BubbleUPnP UpnP/DLNA 软件中，通过 Media 就会显示出手机里面所有的多媒体文件，选定任何一个，树莓派机顶盒就会播放出这个文件，如图 7-52 所示。



图 7-52　在树莓派机顶盒播放 Android 手机中的多媒体影音

4. 教学影片

完整的教学视频可以看 Raspberrypi_XBMC_UPnPAndroid1。

7.8.4　树莓派支持 iOS 操作系统 Airplay

案例 43

Airplay

在 XBMC 部分安装完毕后，默认的功能就已经包含 Airplay 播放，如图 7-53 所示。但是在 iOS 7.x 版本之后，当前 XBMC 的 Airplay 也许是新格式的关系，现在不能正常工作，期待新的 OpenELEC 版本，把这个问题解决掉。如果有旧版本的 iOS，就可以使用这功能来享受在大屏幕上观看 iOS 的影片。

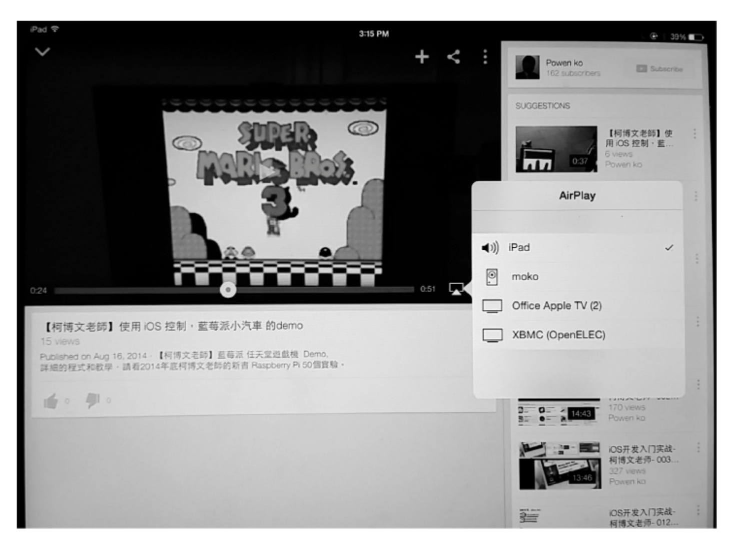


图 7-53　XBMC 支持 Airplay

7.8.5　播放树莓派机顶盒多媒体文件

案例 44

1．目的

播放树莓派机顶盒多媒体文件。

2．步骤

STEP1：下载

在 App Store 下载 ArkMC Lite 版本，如图 7-54 所示。

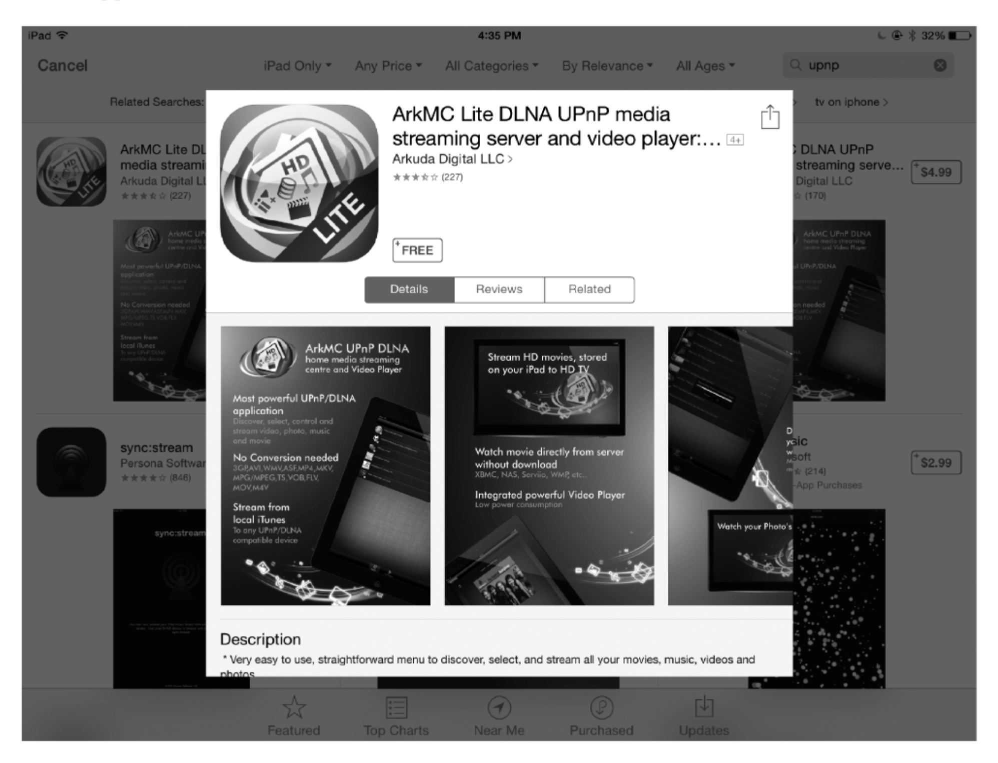


图 7-54　下载 ArkMC Lite 版本

STEP2：打开 ArkMC Lite

在 App Store 下载 ArkMC Lite 之后，如图 7-55 所示，会看到 XBMC（Open ELEC）的选项，单击之后就可以挑选树莓派机顶盒上的多媒体文件。

注意　（1）确认树莓派上的 System→settings→Service→UPnP 项已经勾选。
（2）iOS 和树莓派在同一个局域网内。

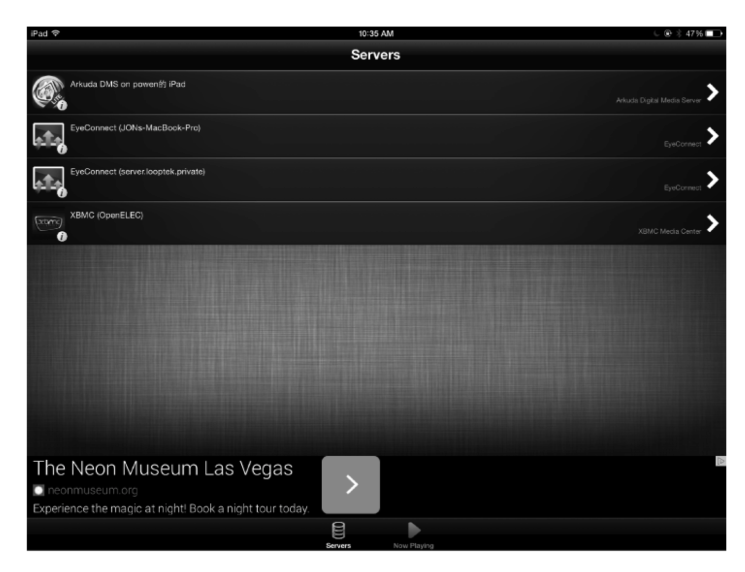


图 7-55　ArkMC Lite

STEP3：播放树莓派上的多媒体文件

在 iOS 挑选文件之后，就可以选择将多媒体文件播放在 iOS 上还是树莓派的屏幕上，如图 7-56 所示。



图 7-56　在 iOS 上播放树莓派的多媒体文件

注意　不是每个多媒体文件都可以在手机上播放，当前 iOS 只能播放 MP4、MOV 等影音格式。

7.8.6　播放 iOS 上的多媒体文件

案例 45

1. 目的

播放 iOS 上的多媒体文件。

2. 步骤

继续案例 4.4 的步骤往下执行。

STEP4：在树莓派上播放 iOS 的多媒体文件

同样，在挑选多媒体文件时，可以选择将此文件播放在 iOS 上还是树莓派的屏幕上。如果选择是 XBMC，多媒体文件就会播放在树莓派的屏幕上，如图 7-57 所示。

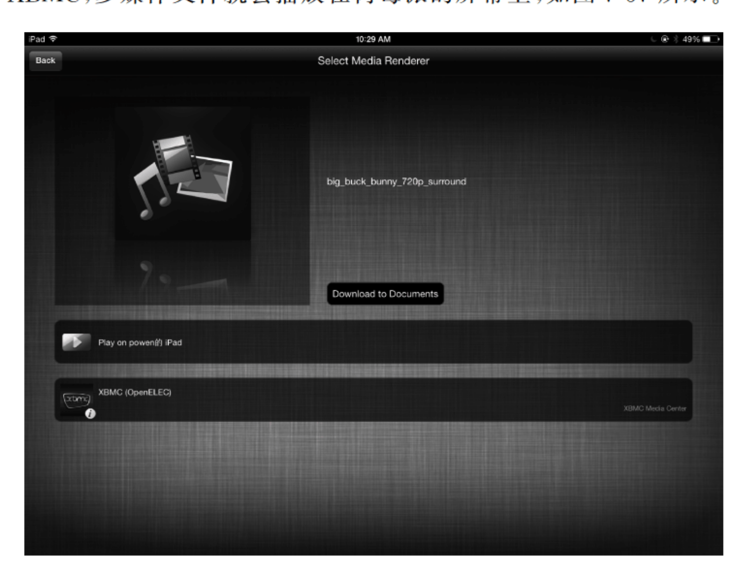


图 7-57　选择播出的设备

STEP5：选择多媒体文件

想将手机上的多媒体播放在树莓派机顶盒的电视屏幕上，请回到指定 Server 的部分，选择 Arkuda DMS，这样就能够把手机上的内容投射到树莓派上。这里 Arkuda Lite 因为是试用版，每个路径只能传输前面 4 个文件。

7.8.7　Windows、Mac 与 Linux 操作系统

案例 46

1. 目的

其他操作系统的操作方法都差不多，下面列出各个操作系统常用的 UPnP Server 的软件。

2. 操作系统

1) Windows 操作系统

```
UPnP Server 软件
  Wild Media Server
  MediaMonkey
  Windows Media Connect
  Universal Media Server
UPnP Player 软件
  MediaMonkey
  WinDVD
  Windows Media Server
  Plex
  XBMC
```

2) MAC 操作系统

```
UPnP Server 软件
  LXiMedia Center
  Universal Media Server

UPnP Player 软件
  MediaColid Mac v2
  SofaPlay
  OPlayer
  VLC Media Player
  Plex
  XBMC
```

3) Linux 操作系统

```
UPnP Server 软件
  Universal Media Server

UPnP Player 软件
  VLC Media Player
  Plex
  XBMC
```

7.9 遥控

7.9.1 iOS 遥控器——Official XBMC Remote

案例 47

1. 介绍

Official XBMC Remote 即可以当成遥控器使用,也可以远程共享树莓派上的多媒体播

放器,如图 7-58 所示,因为设置有些复杂,请按照以下步骤进行操作。



图 7-58　iOS 系统 iPhone 上的 XBMC 的遥控—— Official XBMC Remote

2. 步骤

STEP1：进入设置

在 iOS 的 APP Store 中找到 Official XBMC Remote 这个 APP,如图 7-59 所示,下载并安装到 iPhone 或 iPad 上。

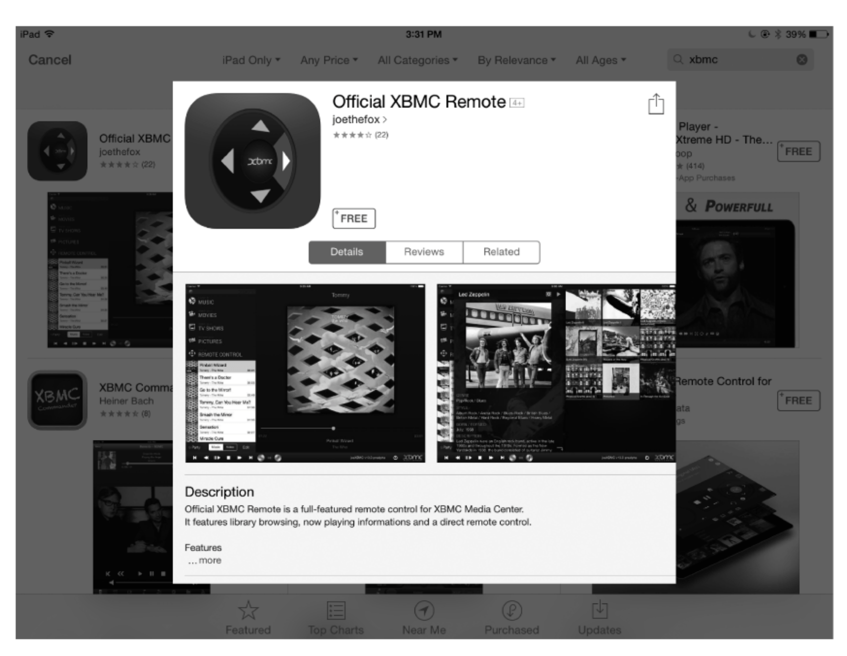


图 7-59　安装 Official XBMC Remote

STEP2：设置准备

首先确认树莓派机顶盒内部的设置：

先进入 System→Services→Web Server，勾选 Allow control of XBMC via HTTP，如图 7-60 所示。



图 7-60　Allow control of XBMC via HTTP 打勾确认

再进入 System→System info→Network，确认树莓派机顶盒的网络地址 IP address 和网卡地址 MAC Address，如图 7-61 所示。

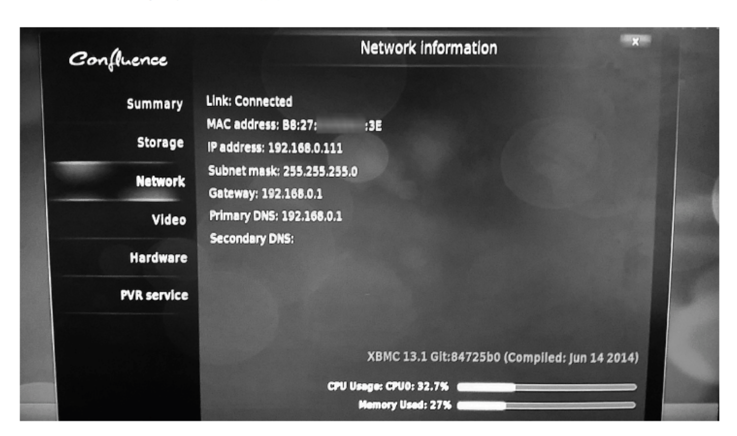


图 7-61　网络信息

STEP3：打开 Official XBMC Remote

在 iOS 设备上把 Official XBMC Remote 打开，第一次打开的时候，APP 会询问和需要以下的连接设置数据，如图 7-62 所示。

Description：树莓派的昵称。



图 7-62　连接设置数据

Host port/TCP port：填写树莓派的网络地址，在后面两位数字分别填上 80/9090。

MAC Address：填写树莓派网络的 MAC 地址。

Username and password：分别填写 root 和对应密码。

单击 Save 按钮之后就可以连接上。

连接成功后就可以用左侧菜单执行系统的功能，或者通过遥控器来遥控操作树莓派机顶盒，执行效果如图 7-63 所示。



图 7-63　iOS 系统 iPad 上的 XBMC 的遥控——Official XBMC Remote

执行后的效果如图 7-64 所示,视频和声音全部都通过树莓派输出到电视机上,iOS 只是做遥控而不执行播放和分享文件的功能。

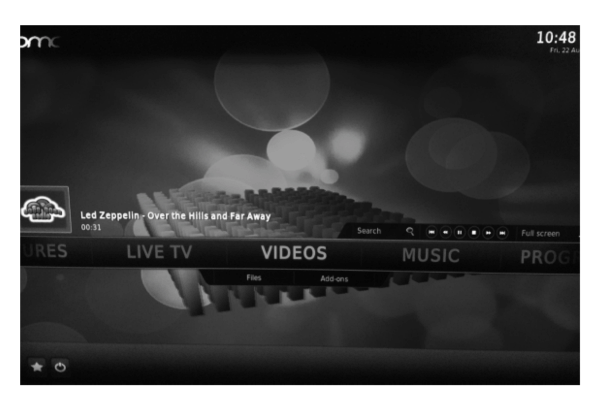


图 7-64 使用 Official XBMC Remote 遥控树莓派机顶盒

7.9.2 Android 遥控器——Yatse 与 Official XBMC Remote

案例 48

这里有两套 Android 系统的 XBMC 的遥控可供选择:

(1) Official XBMC Remote;

(2) Yatse。

通过 Google Play 查找 Yatse,就可以免费获取,如图 7-65 所示。

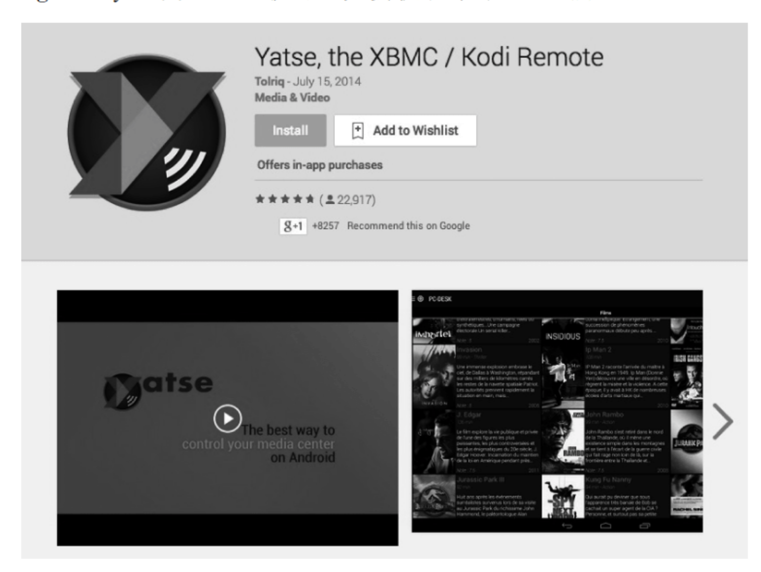


图 7-65 Android 系统的 XBMC 的遥控——Yatse

因为 Android 系统上的 Official XBMC Remote 的操作和设置,跟 iOS 系统上的一样,所以这里就不重复叙述。

7.9.3 USB 遥控器

案例 49

1. 介绍

XBMC 支持非常多的万用遥控器,推荐可以用以下的配备来无线控制树莓派机顶盒。

2. USB 遥控器

1) 标准的 USB 无线键盘

有很多 USB 的无线键盘也可以控制 XBMC,树莓派机顶盒通常只需要上、下、左、右、Enter 和 Esc 键,可以挑选一个体积比较小的 USB 无线键盘,例如如图 7-66 所示,不用设置就能直接使用。

2) FLIRC

FLIRC 是一个 USB 的专用 IR 接收器,如图 7-67 所示,并且可以指定某个信号转换为实体键盘的某一个按键,这样就能把原有的 IR 遥控器设置为树莓派机顶盒遥控器。



图 7-66 Measy RC12 标准的 USB 无线键盘



图 7-67 FLIRC 是 USB 的专用 IR 接收器

7.10 创建 strm 文件

案例 50

1. 目的

网络上有很多实时的 Live TV 和电台等,以下介绍如何在 XBMC 上设置和使用播放。

2. 步骤

STEP1:通过 SSH 进入 XBMC 的字符界面

在计算机上把 SSH 工具打开,通过树莓派的网络地址连接到该设备,这里特别的提醒

OpenELEC 的账号与密码是

OpenELEC 账号：root

密码：OpenELEC

例如 192.168.0.111,连接时使用以下的命令,如图 7-68 所示。

```
$ ssh root@192.168.0.111
```

```
powens-MacBook-Air:Desktop powenko$ ssh root@192.168.0.111
root@192.168.0.111's password:
##############################################
# OpenELEC - The living room PC for everyone #
# ...... visit http://www.openelec.tv ...... #
##############################################

OpenELEC (official) Version: 4.0.5
OpenELEC:~ # ls
lost+found   music        pictures     recordings   screenshots  tvshows      videos
```

图 7-68　通过 SSH 进入 XBMC 的字符界面

STEP2：创建路径

XBMC 只要在 music、pictures、videos 中创建路径和文件,就能在系统上看到对应的路径。

首先进入 videos,通过以下命令创建一个子目录,如图 7-69 所示。

```
$ cd videos
$ mkdir myvideo
```

```
OpenELEC:~/videos # mkdir myvideo
OpenELEC:~/videos # ls
myvideo
```

图 7-69　在 videos 下创建一个子目录

此时在 XBMC 的 Videos 中就会出现对应的路径,如图 7-70 所示。



图 7-70　在 Videos 下出现相同名称的子目录

STEP3：测试文件

为了确保文件可以顺利地播放，可以先用多媒体播放器试试看该网络连接是否正确，例如用 VLC 软件，如图 7-71 所示。



图 7-71　使用 VLC 确认网络连接是否正确

STEP4：创建 strm 文件

在刚刚创建的一个目录下，使用文本编辑器创建一个文件，如图 7-72 所示，注意这个文件名一定要是 strm 后缀，例如

```
$ nano test.strm
```

STEP5：strm 文件内容

把网络连接的地址粘贴到这个文件里，如图 7-73 所示，一次只能一个地址，如果有多条数据，请另外创建 strm 文件。这里的网络连接可以是 mms、RTSP、RTP、HTTP、HTTPS 等，也可以是.PLS 或 .M3U 这类播放清单的网络连接。

```
OpenELEC:~/videos # mkdir myvideo
OpenELEC:~/videos # ls
myvideo
OpenELEC:~/videos # ls
myvideo
OpenELEC:~/videos # cd ..
OpenELEC:~ # cd videos/
OpenELEC:~/videos # cd myvideo/
OpenELEC:~/videos/myvideo # nano test.strm
```

图 7-72　创建 strm 文件

3. 结果

完成之后回到 XBMC 中的 videos，就可以看到该文件播放之后网络上的影片，如果是实时影片，还能够有 PVR 的功能，如图 7-74 所示。

4. 教学影片

完整的教学视频可以看 Raspberrypi_XBMC_CreateTVListing.mov。

注意　多媒体格式非常多种，因为 XBMC 解码器的问题，不是每一种都可以顺利播放，可以多安装几个软件解码器，或者使用前先测试一下，以确保正常播放。



图 7-73　strm 文件格式



图 7-74　执行结果

7.11　看电视、录像和 PVR

案例 51

1. 目的

可以把树莓派机顶盒设置为看电视和录像的设备，如果要录像使用内存大一点的 SD

卡,这样就能多录几部影片。目前为止它支持 NTSC、DVB-T、ASTC 的电视信号,如果想录制高品质的数字电视,必须先设置添加支持 VC-1 和 MEPG-2 解码。

2. 硬件

强烈建议不要乱买 USB TV 棒,以免 OpenELEC 的 Linux 操作系统找不到合适的驱动程序,详细兼容的硬件可以看 OpenELEC 官方介绍的设备,网址为 http://wiki.OpenELEC.tv/index.php/Supported_TV_Tuners,例如 USB TV 棒如图 7-75 所示。



图 7-75　GIGABYTE 的 GT-U7000-RH 的 DVB-T 电视卡

3. 步骤

STEP1:添加 tvheadend Services

需要添加 tvheadend 服务,如图 7-76 所示。

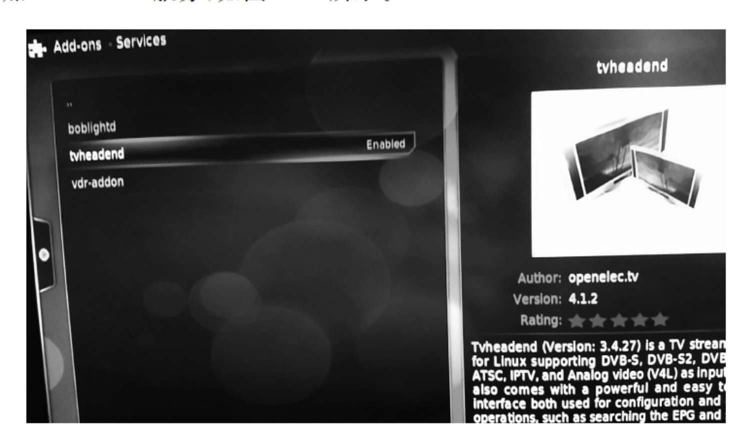


图 7-76　添加 tvheadend Services

进入 Settings→Add-ons→Get add-ons→OpenELEC Mediacenter OS Add-ons→Services 服务,选择 tvheadend,单击 Install 按钮安装 tvheadend,确认 USB 电视棒已经接在树莓派上,

重新引导树莓派。

STEP2：打开 TV 设置

进入 Settings→Live TV→General 选项中，设置为 Enabled，此时在右下角会显示安装设置成功的消息，如图 7-77 所示。

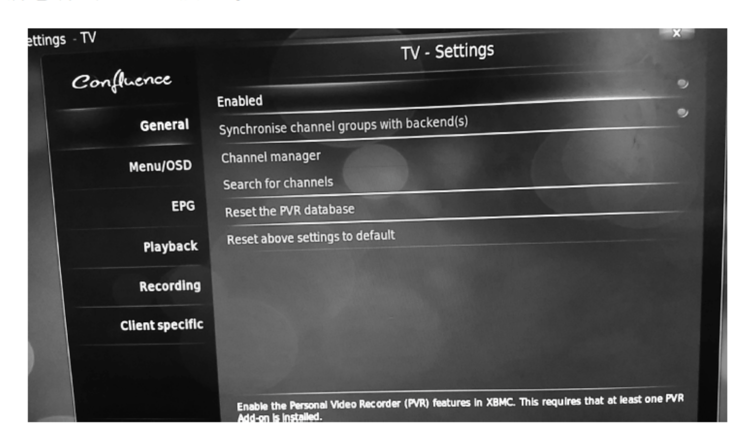


图 7-77　Live TV 设置为 Enabled

STEP3：安装 PVR Client

此时会显示需要安装 PVR Client 的程序，单击 OK 按钮就会出现列表，选择 Tvheadend HTSP Client 项，如图 7-78 所示。再次重新引导树莓派。

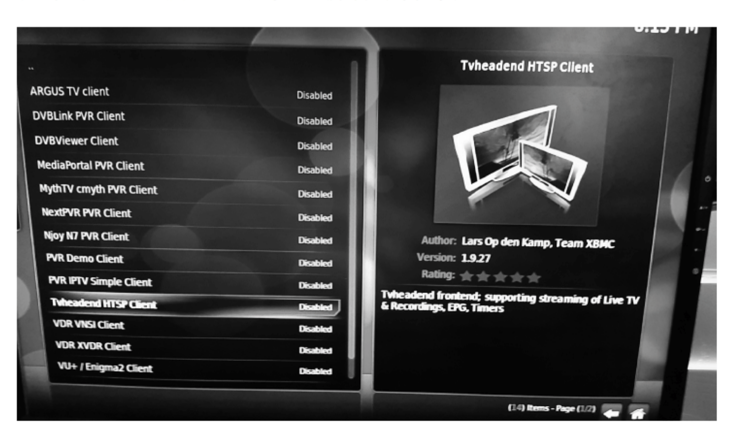


图 7-78　引导 Tvheadend HTSP Client

STEP4：打开 OpenELEC 网址

在浏览器中输入 http://网址：9981，把 OpenELEC 的网址打开，如图 7-79 所示，例如网址为 http://192.168.0.111:9981。



图 7-79　打开 OpenELEC 的网址

STEP5：指定电视卡

进入 Configuration→DVB Inputs→TV Adaptors，通过下拉菜单，选择 TV tuner，打开 Autodetect muxes 和 Idle scanning，单击 Save 按钮，如图 7-80 所示。



图 7-80　指定电视卡

注意 如果没有显示 USB 电视卡名称,请使用以下方法修复:

(1) 需要 USB Hub 供电,因为电视卡需要功率很大,树莓派可能供电不足。

(2) 确定此 USB 电视卡驱动程序顺利安装。

STEP6:指定电视的频率

单击 Add DVB Network by location 按钮,在弹出窗中选择所在地。单击 Add DVB Network 按钮,用来指定电视的频率,如图 7-81 所示。

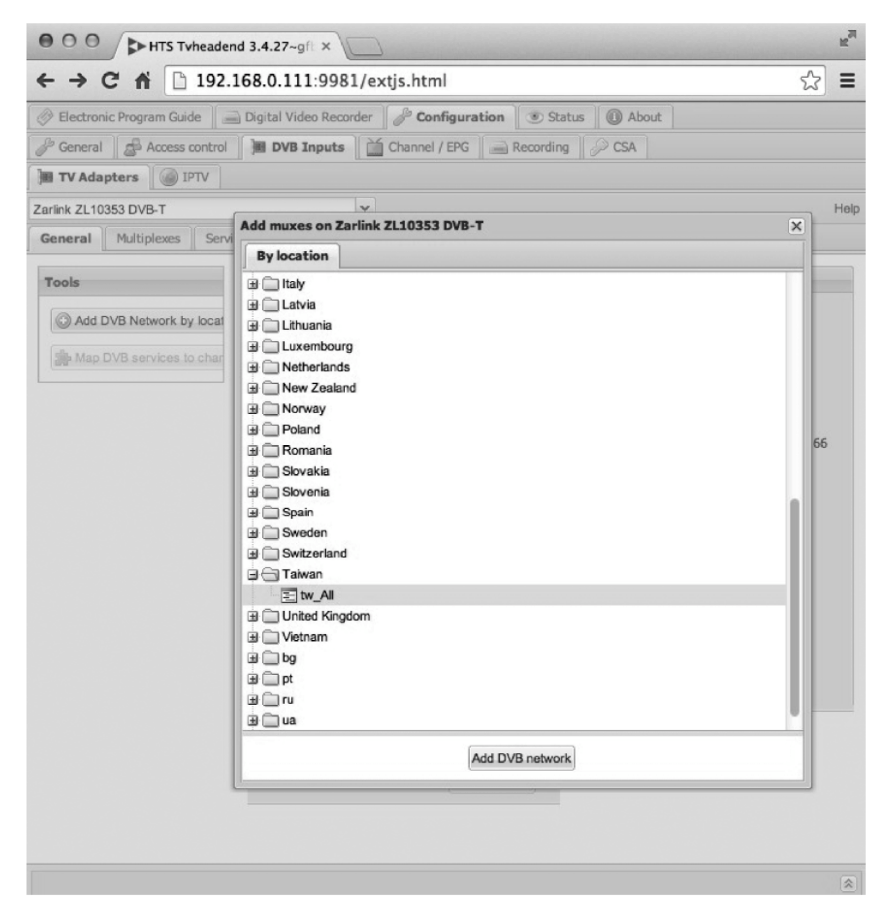


图 7-81 指定电视的频率

STEP7:电视信号的频率

等候 Muxes awaiting initial scan 的数据变成 0。单击 Map DVB services to channels 按钮,如图 7-82 所示。



图 7-82　电视信号的频率

STEP8：电视频道列表

单击 channel/DVB 按钮，就会列出找到的所有电视频道，如图 7-83 所示。

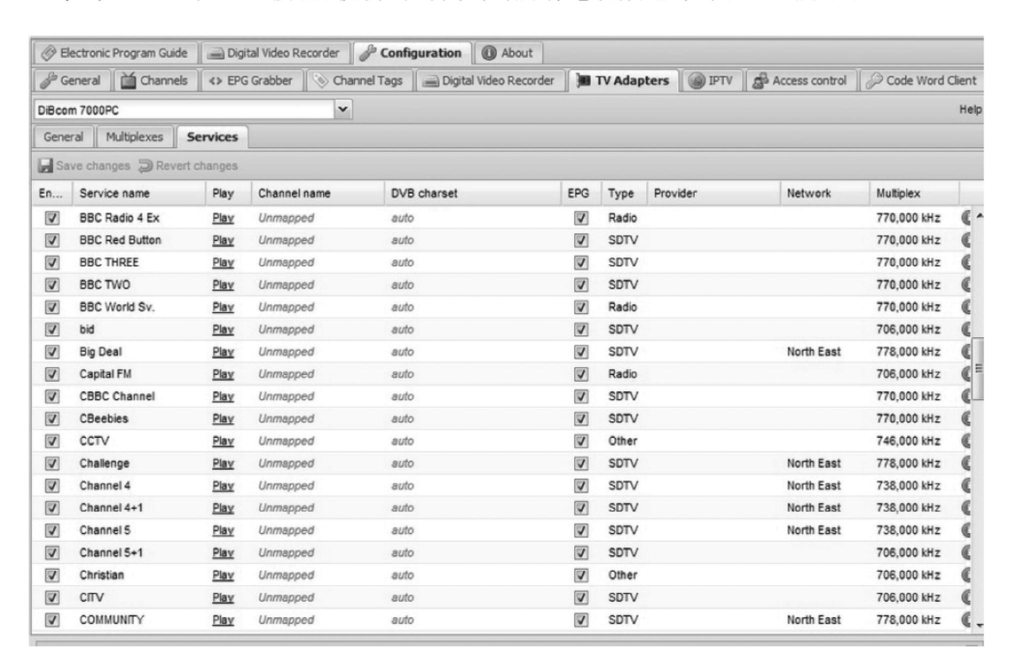


图 7-83　电视频道列表

4. 结果

设置完毕后,回到树莓派的 Live TV 中,如图 7-84 所示,就可以看电视了。



图 7-84　到 LIVE TV 看电视

下面就可以在树莓派上,通过 USB 电视卡看到实时的数字电视,如图 7-85 所示,建议找好的电视信号源,才能看到更多更清楚的电视节目。

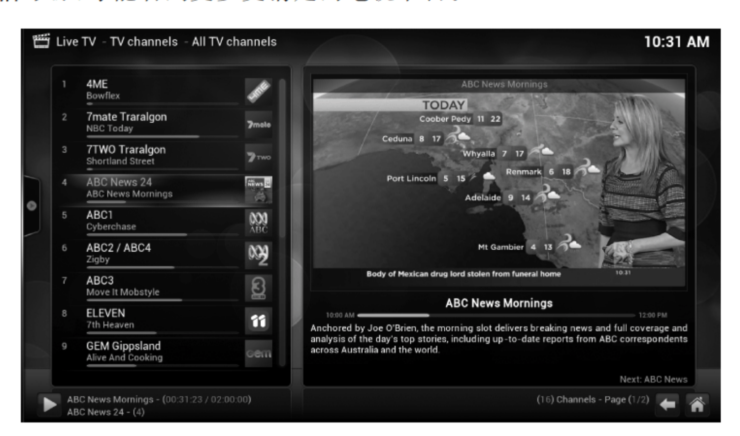


图 7-85　在树莓派上看实时电视

播放出高品质的画面外,还提供了 PVR 的功能,即暂停、倒退、前进、录像等功能,如图 7-86 所示。

由于当前支持的 USB 电视卡数量不多,所以操作时常常会因为驱动程序问题产生麻烦,所以要么找一个官方推荐的电视卡,要么多加练习,学会安装驱动程序。

5. 教学影片

完整的教学视频可以看 Raspberrypi_XBMC_LIVE_TV。



图 7-86　实时电视 PVR

第 8 章 Android 操作系统树莓派

案例 52

1. 介绍

本章介绍如何在树莓派上安装 Android 操作系统,2014 年 Broadcom 虽然发布了树莓派 Android 4.x 版本,并随后暂停了此计划,原因是树莓派使用 Broadcom BCM2835 SoC 芯片和 ARMv6 的 ARM1176JZ-F 处理器,而不是 Android 智能手机所用的 ARMv7,而树莓派 2 已使用 ARMv7,目前已有 Android 4.4 系统版本可下载使用。

目前为止树莓派上可以顺利执行的版本只有 Android 2.3,但该开源代码的版本 Android 系统还算是实验室阶段,因为是使用软件解码,没有使用树莓派上的图形硬件解码,所以执行的速度还需要再做调整。本章介绍如何搭建此版本。

2. 步骤

STEP1:下载 Android 2.3

在浏览器中输入 http://androidpi.wikia.com/wiki/Android_Pi_Wiki,下载树莓派的最新版本即 Android 3.2 版(CyanogenMOD 7.2)的压缩文件,如图 8-1 所示。

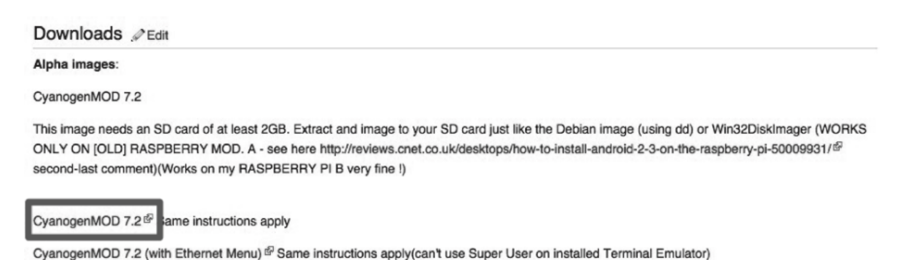


图 8-1　下载 Android 2.3

下载的为.gz 的压缩文件,可以通过 WinZip 软件解压缩。

STEP2:执行 Win32DiskImager 工具

刻录 cm7_img_x.img 的 Android 镜像文件到 SD 卡中。

(1) 把 4GB 以上容量的 SD 卡插入 Windows 系统的计算机。

(2) Windows Vista、Windows 7 和 Windows 8 操作系统的用户,推荐单击鼠标右键,选中 Run as administrator 执行程序 Win32DiskImager. exe,如果是 Windows XP 就直接打开执行。

(3) 指定 img 镜像文件和 SD 卡的位置。

(4) 单击 Write 按钮,如图 8-2 所示。

(5) 如果 SD 卡内有数据,会弹出是否要清除数据的对话窗,单击 Yes。

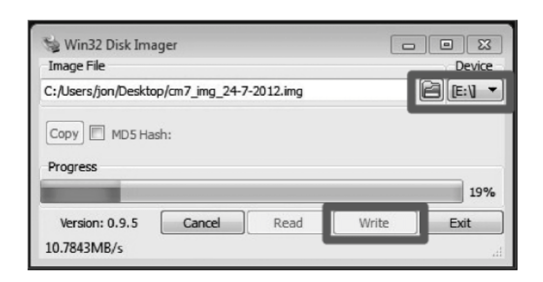


图 8-2 Win32DiskImager 窗

STEP3:执行完成

接下来会看到滚动条由 0%到 100% 的处理过程,完成后出现 Complete 窗口,单击 OK 就可以了。

STEP4:确认

查看 SD 卡的数据是否存在,里面会看到一个 Android 操作系统的文件夹,如图 8-3 所示。确认后把 SD 卡取出来,插在树莓派上。

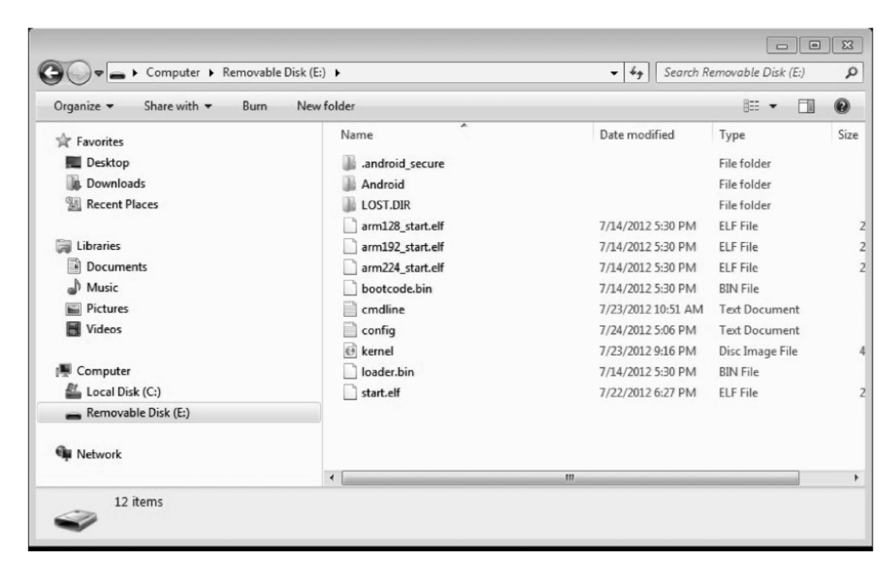


图 8-3 成功后查看 SD 卡的数据是否存在

STEP5：引导 Android 系统树莓派

当第一次把安装过 Android 系统的 img 镜像文件刻录成 SD 卡，并把该 SD 卡放入树莓派，开机之后就能看到完整的 Android 操作系统在树莓派上执行，不过有些 APP 程序执行后会宕掉，稳定性还是存在问题，执行情况如图 8-4 所示。



图 8-4 Android 系统运行在树莓派上

3. 教学影片

完整的教学视频可以看 RaspberryPi_installAndroid。

第 9 章

LCD 点阵式液晶气象预报机

9.1　获取 Yahoo 天气信息

案例 53

1. 介绍

网络上的信息很多,并且有很多 Web Service,如果能通过树莓派取得网络上的服务信息,一定对应用很有帮助,但要如何获得呢?

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络连接设备。

3. 接线

将树莓派连接到网络,并且可以顺利取得网络数据。

4. 步骤

STEP1:确认天气 Web Service 网址

本案例是使用 Yahoo 的 Web Service 来完成,详细的 Web Service 可以查看 http://weather.yahoo.com/,如图 9-1 所示。例如想知道拉斯维加斯的天气,可输入 Las Vegas,将会定向到网址 http://weather.yahoo.com/taiwan/null/taipei-2436704/,知道 w 代码是 2306179,API URL 就会是 http://weather.yahooapis.com/forecastrss?w= 2436704,默认温度单位是华氏,若要显示温度单位,再多加一个参数 u=c,例如 http://weather.yahooapis.com/forecastrss?w=2436704&u=c。

当前 Yahoo 天气的 Web Service 可以通过以下代码切换不同的城市。

1118370 Tokyo(东京)

2151330 Beijing(北京)

2151849 Shanghai(上海)

2306179 Taipei City(台北市)

2436704 Las Vegas(拉斯维加斯)
2459115 New York(纽约)
2306180 Kaohsiung City(高雄市)
2306181 Taichung City(台中市)



图 9-1 Yahoo 天气网站

STEP2：天气 Web Service 的 XML 格式

以拉斯维加斯的天气为例，通过网页连接到 http://weather.yahooapis.com/forecastrss?w=2436704&u=c，就会看到以下天气的相关数据，格式是 XML。如果想获取现在的天气，在 XML 中有一段

```
<yweather:condition text = "Fair" code = "34" temp = "27" date = "Wed,30 Apr 2014 5:52 pm PDT"/>
```

获取一周的天气，可以通过同一个 XML，搜寻

```
<yweather:forecast day = "Wed" date = "30 Apr 2014" low = "14" high = "27" text = "Clear" code
 = "31"/>
<yweather:forecast day = "Thu" date = "1 May 2014" low = "17" high = "29" text = "Sunny" code =
"32"/>
<yweather:forecast day = "Fri" date = "2 May 2014" low = "19" high = "33" text = "Sunny" code =
"32"/>
<yweather:forecast day = "Sat" date = "3 May 2014" low = "21" high = "36" text = "Mostly Sunny"
code = "34"/>
<yweather:forecast day = "Sun" date = "4 May 2014" low = "18" high = "34" text = "Partly Cloudy/
Wind" code = "24"/>
```

就可以看到本周天气预告。

STEP3：安装 Python

安装 Python 程序设计语言之前，先要更新系统的安装程序 apt-get。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

树莓派基本上都安装了 Python，但还是请输入以下命令进行确认，如图 9-2 所示。如果已经安装过，再安装一次也不会影响到系统。

```
$ sudo apt - get install python2.7
```

```
pi@raspberrypi ~ $ sudo apt-get install python2.7
Reading package lists... Done
Building dependency tree
Reading state information... Done
python2.7 is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.
```

图 9-2　安装 Python

STEP4：取得天气 yahoo weather Web Service 的 Python

在树莓派可以通过文字编辑器 nano 编写程序。

```
$ sudo nano yahooweather.py
```

在这个文字编辑器中输入以下程序，并且通过按下 Ctrl+O 组合键和 Enter 键存储程序，和通过 Ctrl+X 组合键和 Enter 键离开文字编辑器 nano。

样例程序：sample\ch9\9-1-yahoo\yahooweather.py

```
1.   #!/usr/bin/python
2.   # create by Powe Ko,www.powenko.com
3.   import sys,getopt
4.   import urllib
5.   from xml.dom import minidom
6.
7.   WEATHER_URL = 'http://xml.weather.yahoo.com/forecastrss?w=%s'
8.   WEATHER_NS = 'http://xml.weather.yahoo.com/ns/rss/1.0'
9.
10.   def weather_for_zip(zip_code):
11.       url = WEATHER_URL % zip_code
12.       dom = minidom.parse(urllib.urlopen(url))
13.       forecasts = []
14.       for node in dom.getElementsByTagNameNS(WEATHER_NS,'forecast'):
15.           forecasts.append({
16.               'date': node.getAttribute('date'),
17.               'low': node.getAttribute('low'),
```

```
18.                 'high': node.getAttribute('high'),
19.                 'condition': node.getAttribute('text')
20.             })
21.         ycondition = dom.getElementsByTagNameNS(WEATHER_NS,'condition')[0]
22.         return {
23.             'current_condition': ycondition.getAttribute('text'),
24.             'current_temp': ycondition.getAttribute('temp'),
25.             'forecasts': forecasts,
26.             'title': dom.getElementsByTagName('title')[0].firstChild.data
27.         }
28.
29.
30.     zipcode = 2436704
31.     if(len(sys.argv)> 0):
32.         zipcode = sys.argv[1]
33.     print weather_for_zip(zipcode)
```

程序解说

第 10 行：定义 Yahoo WebService 函数。

第 12 行：通过 urllib.urlopen 下载网络数据，并且通过 minidom.parse 把文字转换成 XML 格式。

第 14～20 行：通过循环的方法找到 XML 中一周天气的 forecast 关键字。

第 21～27 行：查找 XML 中现在天气的 condition 关键字。

5. 结果

执行以下命令，如图 9-3 所示，便可以获取网络 Yahoo 天气预报的信息，也可以通过修改代码，切换到其他城市。

```
$ python yahooweather.py 2436704
```

```
pi@raspberrypi ~/code/01httpweather $ python yahooweather.py 2436704
2436704
{'current_condition': u'Fair', 'current_temp': u'18', 'title': u'Yahoo! Weather
- Las Vegas, NV', 'forecasts': [{'date': u'1 May 2014', 'high': u'29', 'low': u'
17', 'condition': u'Sunny'}, {'date': u'2 May 2014', 'high': u'33', 'low': u'19'
, 'condition': u'Sunny'}, {'date': u'3 May 2014', 'high': u'36', 'low': u'21', '
condition': u'Mostly Sunny'}, {'date': u'4 May 2014', 'high': u'35', 'low': u'19
', 'condition': u'Partly Cloudy'}, {'date': u'5 May 2014', 'high': u'32', 'low':
 u'16', 'condition': u'Partly Cloudy/Wind'}]}
```

图 9-3　执行结果

6. 教学影片

完整的教学视频可以看 RaspberryPi_01httpweather。

Demo 视频可以看 RaspberryPi_01httpweather_demo。

9.2　使用 16×2 LCD 1602 液晶显示文字

案例 54

1. 介绍

很多情况下，要在树莓派上显示一些简单的数据，这时需要控制 Hitachi HD44780U 芯片，并通过树莓派的 GPIO 传递数据，在 LCD 16×2 显示数字和文字一定很方便。

日立 HD44780 LCD 控制器芯片是最常见的点阵式液晶显示屏(LCD)的显示控制器，如图 9-4 所示，可以连接到一个通用微控制器或微处理器。当前已有如下功能：

(1) 显示 ASCII 字符。

(2) 显示数字。

(3) 显示日文平假名。

(4) 显示一些特殊符号。

(5) 调整显示的亮度。

(6) 芯片最多可以显示 80 个字符，但需要液晶显示屏(LCD)的支持。

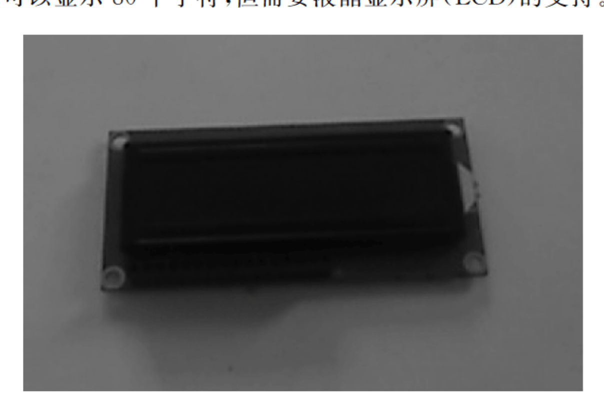


图 9-4　Hitachi HD44780U 的 16×2LCD

2. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 面包板；

(3) 接线若干；

(4) LCD 16×2 Hitachi HD44780U 或者兼容的 LCD 设备；

(5) 可变电阻器。

3. 接线

这里使用的 LCD 2×16 液晶模组是非常常见的 LCD 2×16 液晶模组，注意要用 Hitachi

HD44780U 的芯片控制,如图 9-5 所示。如果不能找到相关的液晶模组,可以到柯博文老师的网站查看清单,http://raspberry.powenko.com。

下面介绍树莓派的引脚用途,如图 9-6 所示。LCD 16×2 Hitachi HD44780U 引脚如何接到树莓派的 GPIO 引脚,可以参考表 9-1,接线如图 9-7 所示。



图 9-5 LCD 16×2 Hitachi HD44780U 引脚

3.3V	1	2	5V
(SDA) *GPIO2	3	4	5V
(SCL) *GPIO3	5	6	GROUND
GPIO4	7	8	GPIO14 (TxD)
GROUND	9	10	GPIO15 (RxD)
GPIO17	11	12	GPIO18
*GPIO27	13	14	GROUND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
(MOSI) GPIO10	19	20	GROUND
(MISO) GPIO9	21	22	GPIO25
(SCKL) GPIO11	23	24	GPIO8 (CE0)
GROUND	25	26	GPIO7 (CE1)

图 9-6 树莓派 GPIO 的引脚功能示意

表 9-1 硬件接线

LCD 2×16 面板的引脚	说明	Pi 引脚说明	Pi 的引脚
01	GND	GND	引脚 06
02	+5V	+5V	引脚 02
03	Contrast	GND	引脚 06
04	RS	GPIO7	引脚 26
05	RW	GND	引脚 06
06	E	GPIO8	引脚 24
07	数据 Bit 0		
08	数据 Bit 1		
09	数据 Bit 2		
10	数据 Bit 3		
11	数据 Bit 4	GPIO25	引脚 22
12	数据 Bit 5	GPIO24	引脚 18
13	数据 Bit 6	GPIO23	引脚 16
14	数据 Bit 7	GPIO18	引脚 12
15	+5V 通过 560Ω 的电阻		
16	接地 GND		GPIO 引脚 06

4. 步骤

STEP1:LCD 显示

完成硬件接线后,执行以下程序,便可以在 LCD 上显示出文字,如果屏幕过亮或看不清楚文字,调整可变电阻就可以改变 LCD 的亮度。

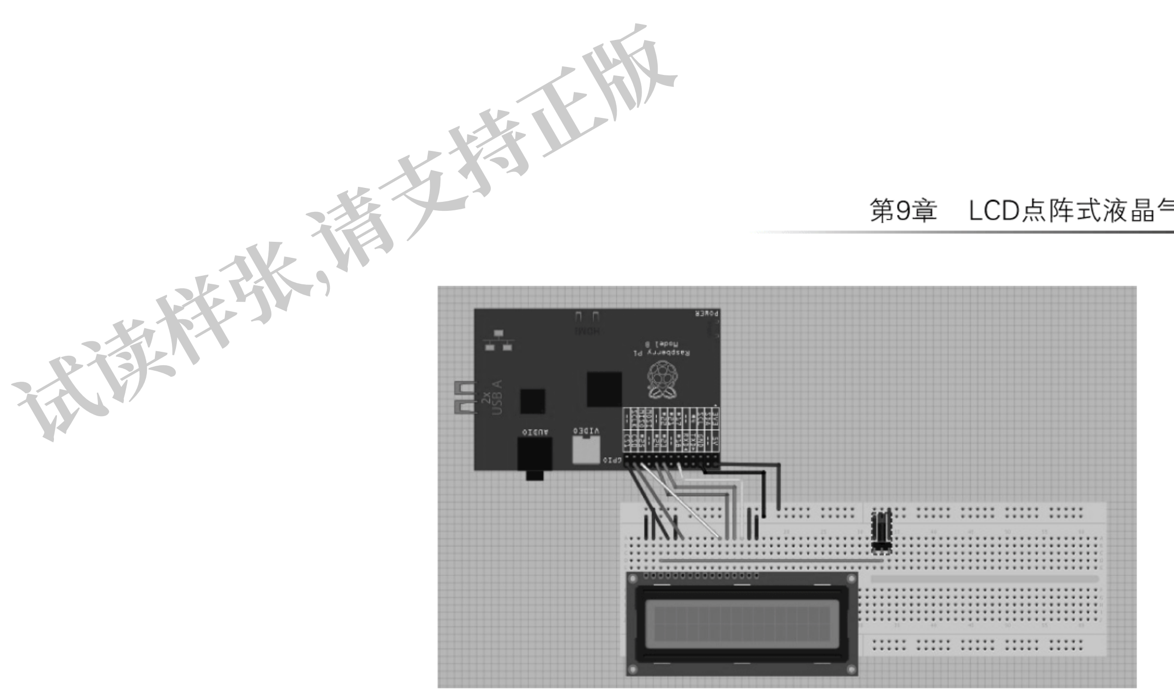


图 9-7　接线图

样例程序：sample\ch9\9-2-lcd\lcdv2.py

```
1.  #!/usr/bin/python
2.  # create by Thetechnoman
3.  # update by Powen Ko,www.powenko.com
4.  import RPi.GPIO as GPIO
5.  import sys
6.  import time
7.
8.  LCD_RS = 7                          # 定义树莓树的 GPIO 对应到 LCD 引脚
9.  LCD_E = 8
10. LCD_D4 = 25
11. LCD_D5 = 24
12. LCD_D6 = 23
13. LCD_D7 = 18
14. LCD_WIDTH = 16                      # 一排最多容纳多少字
15. LCD_CHR = True
16. LCD_CMD = False
17. LCD_LINE_1 = 0x80                   # LCD 第 1 排的内存地址
18. LCD_LINE_2 = 0xC0                   # LCD 第 2 排的内存地址
19. E_PULSE = 0.00005                   # 时间常量
20. E_DELAY = 0.00005
21.
22. def main(str1,str2):
23.   GPIO.setwarnings(False)
24.   GPIO.setmode(GPIO.BCM)            # 设置树莓树的 GPIO 用 BCM 的号码来定义
25.   GPIO.setup(LCD_E,GPIO.OUT)        # E
26.   GPIO.setup(LCD_RS,GPIO.OUT)       # RS
27.   GPIO.setup(LCD_D4,GPIO.OUT)       # DB4
```

```
28.   GPIO.setup(LCD_D5,GPIO.OUT)                  # DB5
29.   GPIO.setup(LCD_D6,GPIO.OUT)                  # DB6
30.   GPIO.setup(LCD_D7,GPIO.OUT)                  # DB7
31.   lcd_init()                                   # LCD 设置初始化
32.   lcd_byte(LCD_LINE_1,LCD_CMD)
33.   lcd_string(str1)
34.   lcd_byte(LCD_LINE_2,LCD_CMD)
35.   lcd_string (str2)
36.   time.sleep(1)
37.
38. def lcd_init():
39.   lcd_byte(0x33,LCD_CMD)                       # LCD 设置初始化
40.   lcd_byte(0x32,LCD_CMD)
41.   lcd_byte(0x28,LCD_CMD)
42.   lcd_byte(0x0C,LCD_CMD)
43.   lcd_byte(0x06,LCD_CMD)
44.   lcd_byte(0x01,LCD_CMD)
45.
46. def lcd_string(message):                       # 输送字符串到 LCD 显示屏
47.   message = message.ljust(LCD_WIDTH," ")
48.   for i in range(LCD_WIDTH):
49.       lcd_byte(ord(message[i]),LCD_CHR)
50.
51. def lcd_byte(bits,mode):
52.   GPIO.output(LCD_RS,mode)                     # RS
53.   GPIO.output(LCD_D4,False)                    # 高位
54.   GPIO.output(LCD_D5,False)
55.   GPIO.output(LCD_D6,False)
56.   GPIO.output(LCD_D7,False)
57.   if bits&0x10 == 0x10:
58.       GPIO.output(LCD_D4,True)
59.   if bits&0x20 == 0x20:
60.       GPIO.output(LCD_D5,True)
61.   if bits&0x40 == 0x40:
62.       GPIO.output(LCD_D6,True)
63.   if bits&0x80 == 0x80:
64.       GPIO.output(LCD_D7,True)
65.   # Toggle 'Enable' pin
66.   time.sleep(E_DELAY)
67.   GPIO.output(LCD_E,True)
68.   time.sleep(E_PULSE)
69.   GPIO.output(LCD_E,False)
70.   time.sleep(E_DELAY)
71.   GPIO.output(LCD_D4,False)                    # 低位
72.   GPIO.output(LCD_D5,False)
73.   GPIO.output(LCD_D6,False)
74.   GPIO.output(LCD_D7,False)
75.   if bits&0x01 == 0x01:
76.       GPIO.output(LCD_D4,True)
```

```
77.   if bits&0x02 == 0x02:
78.       GPIO.output(LCD_D5,True)
79.   if bits&0x04 == 0x04:
80.       GPIO.output(LCD_D6,True)
81.   if bits&0x08 == 0x08:
82.       GPIO.output(LCD_D7,True)
83.   time.sleep(E_DELAY)                                    # 触发 'Enable'引脚
84.   GPIO.output(LCD_E,True)
85.   time.sleep(E_PULSE)
86.   GPIO.output(LCD_E,False)
87.   time.sleep(E_DELAY)
88. st = "www.powenko.com " + time.strftime("%H:%M:%S")     #显示时间
89. if len(sys.argv)>1:                                      # 打开程序时,是否有带参数
90.         st = sys.argv[1]                                 #如果有,参数是什么?
91. st += '' * 32
92. str1 = st[:16]
93. str2 = st[16:32]
94. main(str1,str2)
```

程序解说

第 23 行：把警告消息关闭。

第 25～30 行：设置输出的引脚。

第 39～44 行：LCD 设置初始化。

第 51～87 行：设置 LCD 上的文字。

第 88～93 行：处理在打开程序时，是否有带参数。

5. 结果

通过以下命令并可以输入参数显示消息在 LCD 上面，如果没有带参数，便会显示现在时间，执行结果如图 9-8 所示。

```
$ sudo python ./lcdv2.py  "hello powenko.com"
```



图 9-8　执行结果

6. **教学影片**

完整的教学视频可以看 RaspverryPi_LCD16x2。

9.3 在 LCD 显示天气预报

案例 55

1. **介绍**

上一个案例,获取 yahoo 天气并把天气显示在 LCD 16×2 上面。可以自行修改,把网络上的信息通过 Web Service 显示到 LCD 上。

2. **步骤**

STEP1: 硬件接线

先完成案例 54,确认可以将文字显示在 LCD 16×2 的面板上,再进行本案例的步骤。

STEP2: 修改 Web Service 的程序

以下程序跟之前的 yahoo 天气几乎是一样的,不同地方是通过 os. system 的方法,把取得的天气数据,在文本模式执行 python. /lcdv2. py 'xxxxxx'调用另外一个程序,将文字数据输出 LCD 显示。

样例程序: sample\ch9\9-2-lcd\lcdv2. py

```
1.  #!/usr/bin/python
2.  # create by Powen Ko,www.powenko.com
3.  import sys,getopt
4.  import urllib
5.  import os
6.  from xml.dom import minidom
7.
8.  WEATHER_URL = 'http://xml.weather.yahoo.com/forecastrss?u=c&w=%s'
9.  WEATHER_NS = 'http://xml.weather.yahoo.com/ns/rss/1.0'
10.
11. def weather_for_zip(zip_code):
12.     url = WEATHER_URL % zip_code
13.     dom = minidom.parse(urllib.urlopen(url))
14.     forecasts = ""
15.     ycondition = dom.getElementsByTagNameNS(WEATHER_NS,'condition')[0]
16.     n= ycondition.getAttribute('text')
17.     t= ycondition.getAttribute('temp')
18.     return n+" "+t+"C"
19.
20. zipcode=2436704
21. if(len(sys.argv)>1):
22.     zipcode=sys.argv[1]
```

```
23.
24. str1 = weather_for_zip(zipcode)
25. print str1
26. str2 = 'python ./lcdv2.py'
27. newstr = str2 + "'" + str1 + "'"
28. print newstr
29. os.system(newstr)
30.
```

程序解说

第 24 行：获取天气的数据。

第 29 行：调用另外一个程序 python ./lcdv2.py 'xxxxxx'将文字数据输出到 LCD 上。

3. 结果

通过以下命令并可以输入参数显示消息在 Yahoo 天气在 LCD 上面，如图如 9-9 所示，如果没有带参数，便会显示现在 Las Vegas 的天气。

```
$ sudo python ./YahooWeatherOnLCD.py 2436704
```

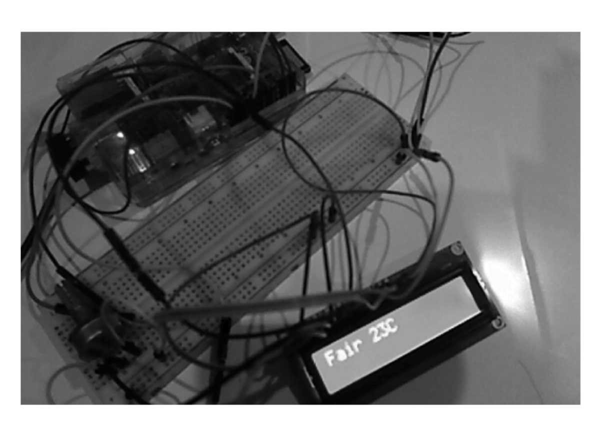


图 9-9 执行结果

4. 教学影片

完整的教学视频可以看 RaspberryPi_WeatherLCD.mov。

第 10 章 物联网温度监测器

现在的天气异常和极端不稳定,使得 40℃以上的高温经常出现,为了保护大自然,可以应用树莓派知识保护我们的自然环境,所以本章最终目的是做出一个监测天气温度,并且定时记录的一个气候监测系统。

10.1 树莓派的 GPIO 模拟输入

树莓派不能直接输入所谓的模拟电压,例如 1.2V 或 0.8V 等,所以需要通过 ADC (Analog to Digital Converters,模拟数字转换器) IC 来读入模拟数据,并通过 IC 转换成 8b 的数字数据,ADC 是一种将连续电压转换成数字的芯片。

常见的 ADC IC 有

ADC0804:有 8 只引脚的数字输出,如图 10-1 所示。

MCP3008:只有 4 只引脚的数字输出。

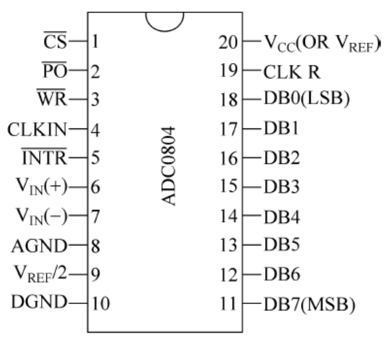


图 10-1 引脚功能

10.1.1 ADC 0804 读入模拟输入——用二进制显示电阻值简易电表

案例 56

1. 目的

因为 ADC 0804 接线已经很复杂了,所以把这次的案例分成两次执行,先来测试 ADC 的 IC 是否接的正确。先读入可变电阻的模拟数据,并通过 LED 灯显示出来,这样就是一个标准的读入电阻值。用二进制显示数据的硬件,这个设备也可以当成"二进制显示电阻值简易电表",把要测试的电阻放上面就知道它的电阻值。

2. 设备

需要有以下硬件:

(1) 1 个 10kΩ 的电阻(棕黑橘);
(2) 8 个 LED 灯;
(3) 1 个 ADC0804 的 IC;
(4) 1 个 150pF 的电容;
(5) 1 个可变电阻器;
(6) 1 个面包板;
(7) 电线若干条。

3. 接线

这个案例是单纯的硬件实验,先关闭电源,按照表 10-1 和图 10-2 的硬件接线图,在面包板上连接硬件线路。

表 10-1 引脚功能

ADC0804 引脚 pin	组 件 引 脚
Pin 1	电池 GND
Pin 2	电池 GND
Pin 3	Pin 5
Pin 4	接到电容
Pin 5	Pin 3
Pin 6	接到可变电阻器
Pin 7	电池 GND
Pin 8	电池 GND
Pin 9	不用接线
Pin 10	电池 GND
Pin 11	第 1 个 LED 灯的长脚,然后 LED 灯的短脚,先接 10kΩ 的电阻器,再接到 GND
Pin 12	第 2 个 LED 灯的长脚,然后 LED 灯的短脚,先接 10kΩ 的电阻器,再接到 GND
Pin 13	第 3 个 LED 灯的长脚,然后 LED 灯的短脚,先接 10kΩ 的电阻器,再接到 GND
Pin 14	第 4 个 LED 灯的长脚,然后 LED 灯的短脚,先接 10kΩ 的电阻器,再接到 GND
Pin 15	第 5 个 LED 灯的长脚,然后 LED 灯的短脚,先接 10kΩ 的电阻器,再接到 GND
Pin 16	第 6 个 LED 灯的长脚,然后 LED 灯的短脚,先接 10kΩ 的电阻器,再接到 GND
Pin 17	第 7 个 LED 灯的长脚,然后 LED 灯的短脚,先接 10kΩ 的电阻器,再接到 GND
Pin 18	第 8 个 LED 灯的长脚,然后 LED 灯的短脚,先接 10kΩ 的电阻器,再接到 GND
Pin 19	先接 10kΩ 的电阻器,再接到电容
Pin 20	3.3V 电池的正极

4. 结果

确定硬件接线正确,再接上 5V 或 3.3V 的电池(两个 AA 的电池),目的是通过调整可变电阻器,可以看到 8 个 LED 灯显示二进制的电阻值。如果手边没有电池,也可以用树莓

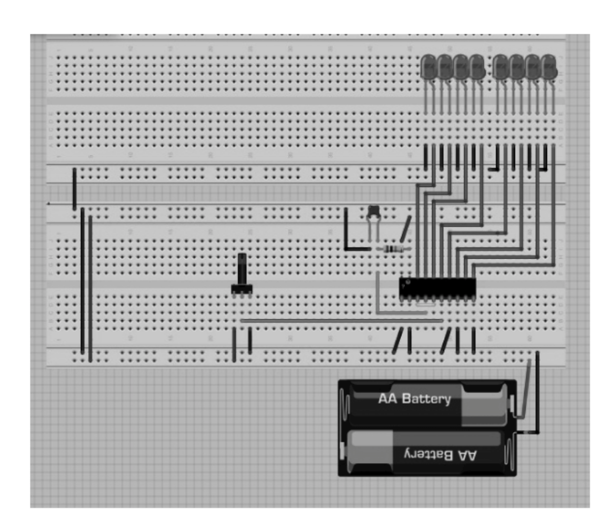


图 10-2　硬件接线设计

派的 GPIO 上的 Pin1 3.3 V 和 Pin6 GND 来代替电池。

取得数字后要如何计算呢?

实际电压=((电池的电压值×读到的数值)/(2^8))

5. 教学影片

因为这个硬件线路比较多,如果您操作不顺利,可以看影片 adc0804 demo,比较一下看哪个引脚没有接好。这个视频中说明如何通过 ADC0804 的 IC 去读入可变电阻,并会把数据值通过 8 个 LED 显示在上面,如图 10-3 所示。

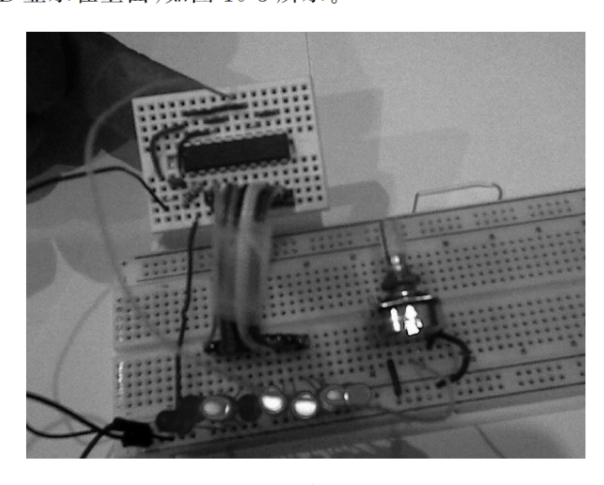


图 10-3　硬件测量结果

10.1.2　树莓派读入模拟数据

案例 57

1. 介绍

本案例的目的是让树莓派读入模拟数据，请先完成实例 56，再继续以下操作，并把这 8 条由 ADC 转换后的数字的数据接到树莓派的 GPIO 引脚上，通过 Python 程序来处理和显示在计算机屏幕上。按照案例 56 先把 ADC0804 接好，关机后拔掉电源，并把案例 56 的电池先拔掉，按照图 10-4 的硬件接线，接上树莓派的硬件线路。改变的是把 LED 灯上面的 8 条接线，接到树莓派的 GPIO 上面，和电源做些调整就可以了。

2. 设备

和案例 56 类似的硬件接线，并把电池拔掉，用树莓派来供电，注意树莓派的 GPIO 有两种电源，分别是 5V 和 3.3 V 直流电，请接在 5V 直流电源，接线图查看表 10-2 和图 10-4 所示。

表 10-2　ADC0804 接线

树莓派引脚 pin	ADC0804 组件
Pin 2 / 电源 5V	Pin 20
Pin 6 / GND	电源 GND
Pin 7 / GPIO4	Pin 18
Pin 11 / GPIO17	Pin 17
Pin 12 / GPIO18	Pin 16
Pin 13 / GPIO21	Pin 15
Pin 15 / GPIO22	Pin 14
Pin 16 / GPIO23	Pin 13
Pin 18 / GPIO24	Pin 12
Pin 22 / GPIO25	Pin 11

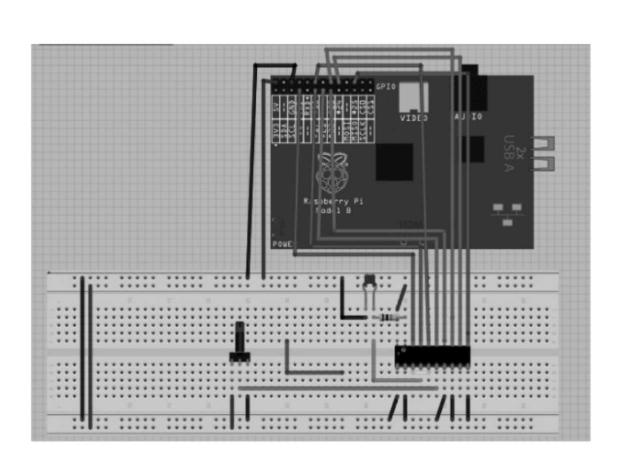


图 10-4　硬件接线

需要有以下硬件：

(1) 1个树莓派；

(2) 电线若干条。

3. 步骤

在文本编辑器中输入以下程序，并保存为analogInput.py。

```
$ nano analogInput.py
```

样例程序：sample\ch10\analogInput.py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
3.  import time,RPi.GPIO as GPIO
4.  GPIO.setmode(GPIO.BOARD)
5.  GPIO.setup(7,GPIO.IN)
6.  GPIO.setup(11,GPIO.IN)
7.  GPIO.setup(12,GPIO.IN)
8.  GPIO.setup(13,GPIO.IN)
9.  GPIO.setup(15,GPIO.IN)
10. GPIO.setup(16,GPIO.IN)
11. GPIO.setup(18,GPIO.IN)
12. GPIO.setup(22,GPIO.IN)
13. while True:
14.         a0 = GPIO.input(7)
15.         a1 = GPIO.input(11)
16.         a2 = GPIO.input(12)
17.         a3 = GPIO.input(13)
18.         a4 = GPIO.input(15)
19.         a5 = GPIO.input(16)
20.         a6 = GPIO.input(18)
21.         a7 = GPIO.input(22)
22.        total = a0 + (a1 * 2) + (a2 * 4) + (a3 * 8) + (a4 * 16) + (a5 * 32) + (a6 * 64) + (a7 * 128)
23.          print a7,a6,a5,a4,a3,a2,a1,a0,"[",total,"]"
24.         time.sleep(0.1)
```

程序解说

第1行：执行环境的定义文档，Python程序存放路径。

第2行：注释的写法，前面加个#就成为注释。

第4行：注意是定义GPIO.BOARD，所以是用实际的引脚号码来做设置。

第5行：设置GPIO 4/Pin 7的引脚，负责数据输入GPIO.IN。

第14行：读入GPIO 4/Pin 7的引脚，存放在变量a0中。

第22行：因为每一个引脚的数据只有1和0，所以换算二进制到十进制以方便阅读。

第23行：显示每个引脚的数据输入和全部的十进制数据。

第 13 行：无尽循环。

输入完成后，通过 Ctrl+O 组合键存储程序，Ctrl+X 组合键离开 nano 编辑器。

4. 结果

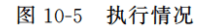
输入以下命令执行 Python 程序，执行效果如图 10-5 所示。要离开时可以按下 Ctrl+Z 组合键强制离开。

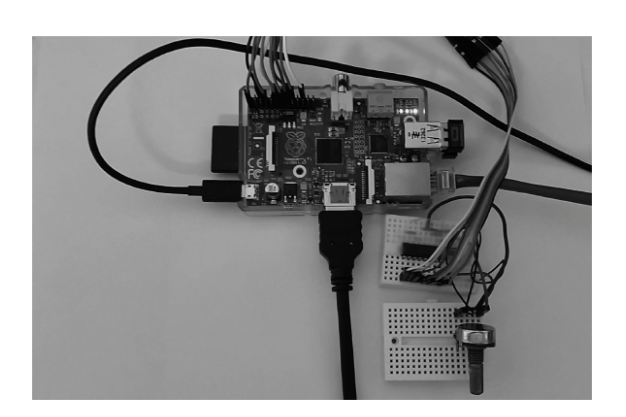
```
$ sudo python analogInput.py
```

5. 教学影片

完整的教学视频可以看 python-gpio-analog-input，该视频是柯博文老师的教学影片，介绍如何通过 Python 去读入可变电阻值，如图 10-6 所示。

```
0 1 1 1 0 0 0 0 [ 112 ]
0 1 1 1 1 1 1 1 [ 127 ]
0 1 1 1 0 0 1 0 [ 114 ]
0 1 1 1 0 0 1 0 [ 114 ]
0 1 1 0 0 1 0 0 [ 100 ]
0 1 0 1 1 0 1 1 [ 91 ]
0 1 0 1 0 0 1 1 [ 83 ]
0 1 0 0 1 1 1 0 [ 78 ]
0 1 0 0 1 0 1 1 [ 75 ]
0 1 0 0 1 0 0 1 [ 73 ]
0 1 0 0 1 0 0 1 [ 73 ]
0 1 0 0 1 0 1 1 [ 75 ]
0 1 0 0 0 1 1 0 [ 70 ]
0 0 1 1 1 1 1 1 [ 63 ]
0 0 1 1 1 1 0 0 [ 60 ]
0 0 1 1 1 1 0 1 [ 61 ]
0 0 1 1 1 1 0 1 [ 61 ]
0 1 0 0 1 0 1 0 [ 74 ]
0 1 1 1 0 0 1 1 [ 115 ]
1 0 0 0 0 0 1 0 [ 130 ]
```


图 10-5 执行情况


图 10-6 实际硬件

注意 如果担心占用了 8 个 GPIO 的引脚后，就没有其他引脚可以使用，还可以使用以下的方法：

(1) 通过 74HC595 这一类的 IC 让树莓派的 3 个 GPIO 的引脚来控制 8 个接脚。

(2) 如果对模拟输入的数据要求不那么准确，可以把 ADC0804 最低的几位删除，D7、D6、D5、D4…这样由低到高以节省引脚。

(3) 通过其他 GPIO 来使用。

---

① 引脚 3 可以当成 GPIO0；

② 引脚 5 可以当成 GPIO1；

③ 引脚 19 可以当成 GPIO10；

④ 引脚 21 可以当成 GPIO9；

⑤ 引脚 23 可以当成 GPIO11；

⑥ 引脚 24 可以当成 GPIO8；

⑦ 引脚 26 可以当成 GPIO7。

10.2 使用温度传感器监测室内温度

案例 58

1. 介绍

本案例将会使用温度传感器来监测室内温度,让树莓派成为一个温度计并把相关数据存储和保留到文件中,以做之后的温度变换记录。

温度的表示方法有许多种,较常见的有摄氏温度、华氏温度及绝对温度。摄氏温度单位为℃,冰点为0℃,沸点为100℃,此种表示法当前最常用;华氏温度也常见于温度计量,其单位为℉,冰点为32 ℉,沸点为212 ℉,两者间相差180 ℉;国际单位系统(International System of Units,SI)的温度单位为凯氏温度(Kelvin,K),属于热力学的温度单位,故没有度的单位,在凯氏温度里,冰点为273.15K,沸点为373.15K,两点之差与摄氏单位类似为100.00K,因此1K等于1℃。凯氏温度的0K称为绝对零度,该点若换算为摄氏温度为-273.15℃。以下简单列出各温度单位之间的数学转换关系,公式均以摄氏温度作为单位。

(1) ℃=(℉-32)×5/9

(2) ℉=9/5 ×℃+32

(3) K=℃+273.15

能够用来感测温度变化的仪器非常多,测温原理都是借电阻变化、电动势变化或颜色、体积的改变等方式间接表现出温度变化。在感测方式上可大致区分为感测器与被测物直接接触之接触式,与不和被测物接触、利用辐射热间接推测温度的非接触式两种。常用于温度感测的仪器种类繁多,应用原理也不尽相同,常见的包括热敏电阻、电阻式温度感测器、热电偶与感温IC等,使用温度传感器LM 35做测量温度的组件来监测温度的反应,并把结果显示在树莓派的屏幕上。

2. 准备

需要先了解“9.1 获取Yahoo天气信息”中的Python安装。

3. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 1个面包板;

(3) 接线若干;

(4) 1个LCD 16×2 Hitachi HD44780U或者兼容的LCD设备;

(5) 1个可变电阻器。

4. 硬件规范

本案例用的是3个引脚的温度传感器LM 35,如图10-7所示。

D0模拟数据的引脚,黑线为地线,红线为电源线,用户通常会通过微处理器连接温度传感器电源、接地与模拟数据的输出,如图10-8所示。它的3个引脚分别是:

(1) D0 模拟数据的引脚;
(2) GND 接地;
(3) VCC 电源。



图 10-7 LM35 温度传感器的外观

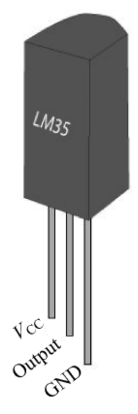


图 10-8 LM35 温度组件

5. 接线

引脚的接线如图 10-9 所示。

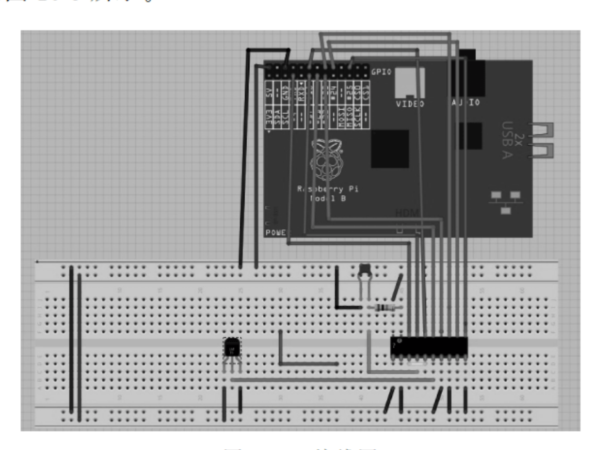


图 10-9 接线图

6. 步骤

STEP1:硬件接线

先完成案例 57,并把可变电阻器换成 LM35 温度组件。

STEP2:编写温度计的程序

通过文本编辑器输入以下程序,并保存成 temperature.py。

```
$ nano temperature.py
```

输入完成后,通过 Ctrl+O 组合键存储程序,Ctrl+X 组合键离开 nano 编辑器。

样例程序:sample\ch10\temperature.py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
3.  import time,RPi.GPIO as GPIO
4.
5.  GPIO.setmode(GPIO.BOARD)
6.  GPIO.setup(7,GPIO.IN)
7.  GPIO.setup(11,GPIO.IN)
8.  GPIO.setup(12,GPIO.IN)
9.  GPIO.setup(13,GPIO.IN)
10. GPIO.setup(15,GPIO.IN)
11. GPIO.setup(16,GPIO.IN)
12. GPIO.setup(18,GPIO.IN)
13. GPIO.setup(22,GPIO.IN)
14. while True:
15.         a0 = GPIO.input(7)
16.         a1 = GPIO.input(11)
17.         a2 = GPIO.input(12)
18.         a3 = GPIO.input(13)
19.         a4 = GPIO.input(15)
20.         a5 = GPIO.input(16)
21.         a6 = GPIO.input(18)
22.         a7 = GPIO.input(22)
23.         total = a0 + (a1 * 2) + (a2 * 4) + (a3 * 8) + (a4 * 16) + (a5 * 32) + (a6 * 64) + (a7 * 128)
24.         temp = total * (5 * 1000/256)/10;
25.         print a7,a6,a5,a4,a3,a2,a1,a0,"[",total,"]","[C = ",temp,"]"
26.         time.sleep(0.1)
```

程序解说

第 1 行:执行环境的定义文档,Python 程序存放路径。

第 2 行:注释的写法,前面加个#就可以成为注释。

第 5 行:注意是定义 GPIO.BOARD,所以是用实际的引脚号码来做设置。

第 6 行:设置 GPIO 4/Pin 7 的引脚,负责数据输入 GPIO.IN。

第 15 行:读入 GPIO 4/Pin 7 的引脚,存放在变量 a0 中。

第 23 行:因为每一个引脚的数据只有 1 和 0,所以换算二进制到十进制以方便阅读。

第 24 行:把模拟数据转换成温度的计算公式。

第 25 行:显示每个引脚的数据输入和全部的十进制数据。

第 14 行:无尽循环。

7. 结果

输入以下命令就能够获取现在的气温,如图 10-10 和图 10-11 所示。

```
$ sudo python temperature.py
```




```
0 0 0 1 0 0 0 1 [ 17 ] [C= 32 ]
0 0 0 1 0 0 0 1 [ 17 ] [C= 32 ]
0 0 0 1 0 0 0 1 [ 17 ] [C= 32 ]
0 0 0 1 0 0 0 1 [ 17 ] [C= 32 ]
0 0 0 1 0 0 0 1 [ 17 ] [C= 32 ]
0 0 0 1 0 0 0 1 [ 17 ] [C= 32 ]
0 0 0 1 0 0 0 1 [ 17 ] [C= 32 ]
0 0 0 1 0 0 0 1 [ 17 ] [C= 32 ]
```

图 10-10　执行结果

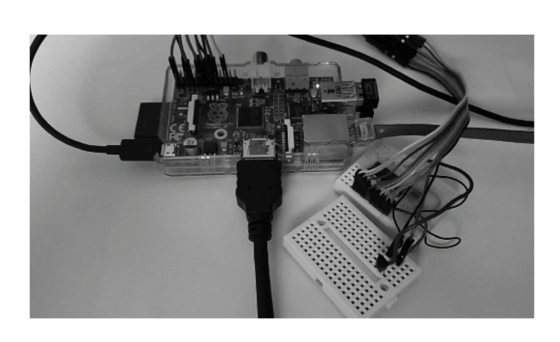


图 10-11　实际硬件图

8. 教学影片

完整的教学视频可以看 RaspberryPi_Temp。

10.3　定时执行程序

案例 59

1. 介绍

本案例介绍如何定时执行指定程序,这样就能够设置树莓派定时去做某些事情和执行特定的程序。

2. 步骤

STEP1:设置执行的操作

首先需要设置 Shell Script,并放在指定路径中,这里通过文本编辑器来设置执行的操作。

```
$ mkdir code
$ cd code
$ sudo nano ./job.sh
```

修改程序把现在的时间写入 1.txt 的文件中,把下面的代码输入到./job.sh 中,存储后再离开,如图 10-12 所示。

```
date >>/home/pi/code/1.txt
```

STEP 2:改变权限

需要修改文件权限,请执行以下命令,如图 10-13 所示。

```
$ sudo chmod 755 job.sh
```

STEP 3:设置执行的时间

输入以下命令来设置 Shell 的执行时间。

```
$ sudo crontab -e
```



图 10-12　code/job.sh 的内容

```
pi@raspberrypi ~/code $ sudo chmod 755 job.sh
pi@raspberrypi ~/code $ ls -al
total 32
drwxr-xr-x  2 pi   pi   4096 Sep  7 21:03 .
drwxr-xr-x 14 pi   pi   4096 Sep  7 20:46 ..
-rw-r--r--  1 root root  174 Sep  7 21:10 1.txt
-rw-r--r--  1 pi   pi    671 Sep  7 16:54 edge.py
-rwxr-xr-x  1 root root   30 Sep  7 20:51 job.sh
-rw-r--r--  1 pi   pi    627 Sep  7 20:07 temperature2.py
-rw-r--r--  1 pi   pi    722 Sep  7 18:51 temperature.py
-rw-r--r--  1 pi   pi   _331 Sep  7 00:47 testspeed.py
```

图 10-13　改变权限

树莓派就会自动打开 nano 文本编辑器在文本最后添加以下命令，如图 10-14 所示。

```
*/1 * * * * /home/pi/code/job.sh
```

意思是每分钟执行/home/pi/code/job.sh 的 Shell Script 一次。完成后，通过 Ctrl+O 组合键存储程序，Ctrl+X 组合键离开编辑器。

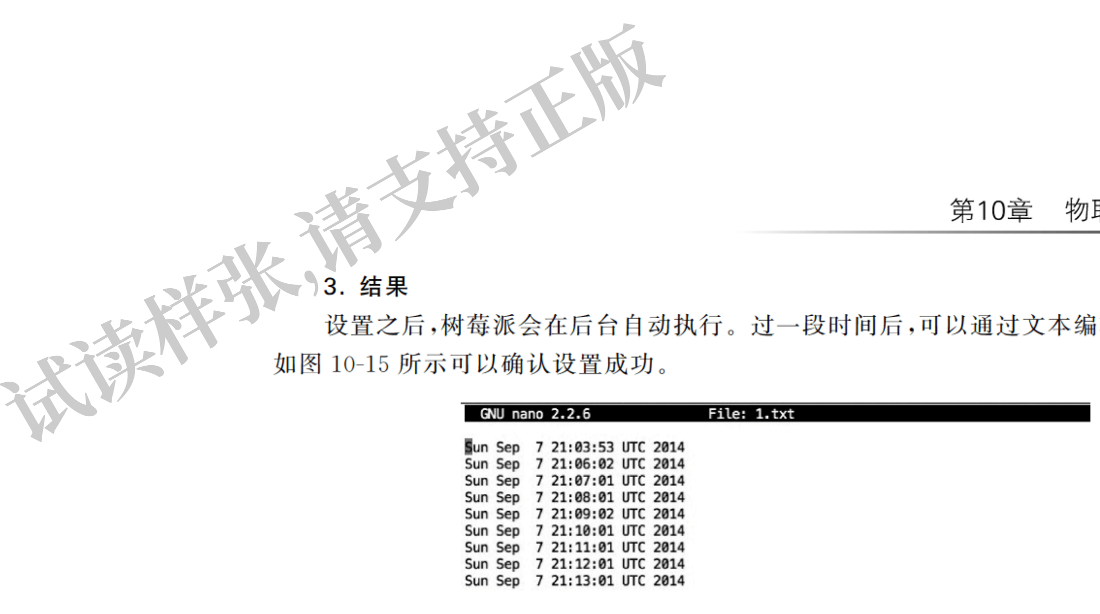
```
GNU nano 2.2.6        File: /tmp/crontab.pTdJkk/crontab              Modified

# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
# 1.M   2.H     3.D     4.M     5.W
#  M = minute  (0 - 59)2. S = hour (0 - 23)3. D = day
#    (1 - 31)4. M = month
#    (1 - 12)5. W=weekday (0 - 7) (Sonntag is represented by 0 and 7)
# m h  dom mon dow   command

*/1 * * * *  /home/pi/code/job.sh

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 10-14　设置执行的时间



3. 结果

设置之后,树莓派会在后台自动执行。过一段时间后,可以通过文本编辑器打开 1. txt,如图 10-15 所示可以确认设置成功。

```
  GNU nano 2.2.6                File: 1.txt

Sun Sep  7 21:03:53 UTC 2014
Sun Sep  7 21:06:02 UTC 2014
Sun Sep  7 21:07:01 UTC 2014
Sun Sep  7 21:08:01 UTC 2014
Sun Sep  7 21:09:02 UTC 2014
Sun Sep  7 21:10:01 UTC 2014
Sun Sep  7 21:11:01 UTC 2014
Sun Sep  7 21:12:01 UTC 2014
Sun Sep  7 21:13:01 UTC 2014
```

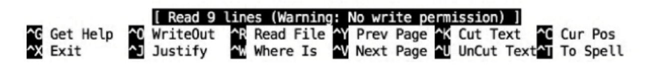


图 10-15　定时执行的结果

4. 教学影片

完整的教学视频可以看 RaspberryPi_CornTab. mov。

注意　在 crontab 中定义执行时间的方法,如下所示。

每 30 分钟执行一次。

```
*/30 * * * * /home/pi/code/job.sh
```

每 5 分钟执行一次。

```
*/5 * * * * /home/pi/code/job.sh
```

上午 8 点 30 分执行一次。

```
30 8 * * * /home/pi/code/job.sh
```

下午 10 点 20 分执行一次。

```
20 22 * * * /home/pi/code/job.sh
```

每月 3 号下午 10 点 20 分执行一次。

```
20 22 3 * * /home/pi/code/job.sh
```

每年 1 月 3 号下午 10 点 20 分执行一次。

```
20 22 3 1 * /home/pi/code/job.sh
```

每周 1 下午 10 点 20 分执行一次,0 是星期天,1 是星期一。

```
20 22 * * 1 /home/pi/code/job.sh
```

10.4 物联网——把数据上传到网络

案例 60

1. 目的

在设计本案例时,在想要用哪一种方法把物联网的数据汇整在一起,是通过网络把数据收集在一起,还是通过 python-mysqldb 模块直接放在本机的数据库中?最后考虑使用网络的方法比较好,可以传递到网络上任何的网站服务器,也可以用 http://localhost 把数据传递给自己本身的服务器。

接收数据端,可以设置在同一台树莓派上或另外一台树莓派上,甚至用 PC 重新搭建。案例 60 介绍如何把数据上传到网页,案例 61 介绍会如何处理接收和存储。

2. 准备

为了完成本案例需要先了解:

(1) 设置固定网络 IP,请查看“3.1 设置固定 IP 网络”。

(2) 搭建网站服务器,请查看“4.1 搭建 PHP 网站服务器”中的 PHP 网站搭建。

(3) 安装 MySQL 数据库,请查看“4.3 创建数据库 MySQL ”中的 MySQL 安装。

(4) 温度计的硬件,请查看“10.2 使用温度传感器监测室内温度”中的相关硬件安装。

3. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 1 个面包板;

(3) 接线若干;

(4) 网络设备。

4. 步骤

首先参考 10.1 节连接好硬件线路。

STEP1:编写温度计的程序

在文本编辑器中输入以下程序,并存成 temp_url.py。

```
$ nano temp_url.py
```

输入完成后,通过 Ctrl+O 组合键存储程序,Ctrl+X 组合键离开 nano 编辑器。

样例程序:sample\ch10\temp_url.py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
```

```
3.  import time,RPi.GPIO as GPIO
4.  import urllib
5.
6.
7.  def fetch_thing(url,params,method):                # 传递数据到网络上的函数
8.      params = urllib.urlencode(params)
9.      if method == 'POST':
10.         f = urllib.urlopen(url,params)
11.     else:
12.         f = urllib.urlopen(url + '?' + params)
13.     return (f.read(),f.code)
14.
15.
16. GPIO.setmode(GPIO.BOARD)                           # 设置树莓派的 GPIO 引脚操作
17. GPIO.setup(7,GPIO.IN)
18. GPIO.setup(11,GPIO.IN)
19. GPIO.setup(12,GPIO.IN)
20. GPIO.setup(13,GPIO.IN)
21. GPIO.setup(15,GPIO.IN)
22. GPIO.setup(16,GPIO.IN)
23. GPIO.setup(18,GPIO.IN)
24. GPIO.setup(22,GPIO.IN)
25.
26. a0 = GPIO.input(7)
27. a1 = GPIO.input(11)
28. a2 = GPIO.input(12)
29. a3 = GPIO.input(13)
30. a4 = GPIO.input(15)
31. a5 = GPIO.input(16)
32. a6 = GPIO.input(18)
33. a7 = GPIO.input(22)
34. total = a0 + (a1 * 2) + (a2 * 4) + (a3 * 8) + (a4 * 16) + (a5 * 32) + (a6 * 64) + (a7 * 128)
35. temp = total * 5 * 1000/256/10;
36. print a7,a6,a5,a4,a3,a2,a1,a0,"[",total,"]","[C = ",temp,"]"
37. content,response_code = fetch_thing(
38.                               'http://127.0.0.1/settemp.php',
39.                               {'id': 1,'temp': temp},
40.                               'GET'
41.                           )
```

程序解说

第 7～13 行：上传网络的函数。

第 37～38 行：把硬件获取的数据，调用自己写的函数上传至网络。

注意　第 38 行有指定网页地址，请按照实际情况自行调整 IP 和网页地址。

5. 结果

输入以下命令,如图 10-16 所示,就能够获取现在的气温,所以可以看到实验室当时的温度是 35℃,并将数据上传至网页,网页的部分案例 61 会有介绍。

```
$ sudo python temp_url.py
```

```
pi@raspberrypi ~/code $ sudo python temp_url.py
0 0 0 1 0 0 1 0 [ 18 ] [C= 35 ]
```

图 10-16 执行结果

6. 教学影片

完整的教学视频可以看 RaspberryPi_Temperature_URL。

10.5 物联网——把数据写到数据库

案例 61

1. 目的

本案例会使用 PHP 程序语言和 MySQL 数据库来存储数据,完成后通过树莓派来收集数据,这样就能够成为一个物联网的数据汇整中心。树莓派数据汇整中心,可以用同一台树莓派来搭建,也可以用另外一台树莓派当成物联网的数据汇整中心。

2. 准备

为了完成本案例需要先了解:

(1) 设置固定网络 IP,请查看“3.1 设置固定 IP 网络”。

(2) 搭建网站服务器,请查看“4.1 搭建 PHP 网站服务器”中的 PHP 网站搭建。

(3) 安装 MySQL 数据库,请查看“4.2 创建数据库 MySQL”中的 MySQL 安装。

(4) 温度计的硬件,请查看“10.2 使用温度传感器监测室内温度”中的相关硬件安装。

3. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 1 个面包板;

(3) 接线若干;

(4) 网络设备。

4. 步骤

首先把网站服务器和 MySQL 数据库服务器搭建好。

STEP1:数据库的创建

在浏览器中输入以下网址,连接到 MySQL 管理界面。

```
http://192.168.0.91/phpmyadmin
```

单击 Database 创建一个新的数据库 RaspberryDB 并使用 utf8_unicode_ci 编码，如图 10-17 所示。



图 10-17　数据库的创建

STEP2：数据库表单的创建

在此数据库中创建一个新的表单，可以通过 SQL 语法执行以下的命令，如图 10-18 所示。

```
CREATE TABLE temp(datetime DATETIME NOT NULL,temp FLOAT NOT null,userid INT NOT null);
```

STEP3：编写 PHP 程序

编写一个 PHP 程序把通过 URL 接收到数据写入数据库中。在文本编辑器中输入以下程序，并存成 settemp. php。

```
$ sudo nano /var/www/settemp.php
```

输入完成后，通过 Ctrl+O 组合键存储程序，Ctrl+X 组合键离开 nano 编辑器，如图 10-19 所示。

样例程序：sample\ch10\settemp. php

```
1.  <?php
2.   $con = mysqli_connect("localhost","root","raspberry","raspberryDB");
```

```
3.  if (mysqli_connect_errno()) {
4.   echo "Failed to connect to MySQL: " .  mysqli_connect_error();
5.  }
6.
7.
8.   $ now1 = date('Ymdhis');
9.   $ id =  $ _GET['id'];
10.  $ temp =  $ _GET['temp'];
11. mysqli_query( $ con,"INSERT INTO temp(datetime,temp,userid)
12. VALUES( $ now1, $ temp, $ id)");
13.
14. mysqli_close( $ con);
15. echo "powenko.com get it".",date time = ". $ now.",temp = ". $ temp."  id = ". $ id;
16. ?>
17.
```

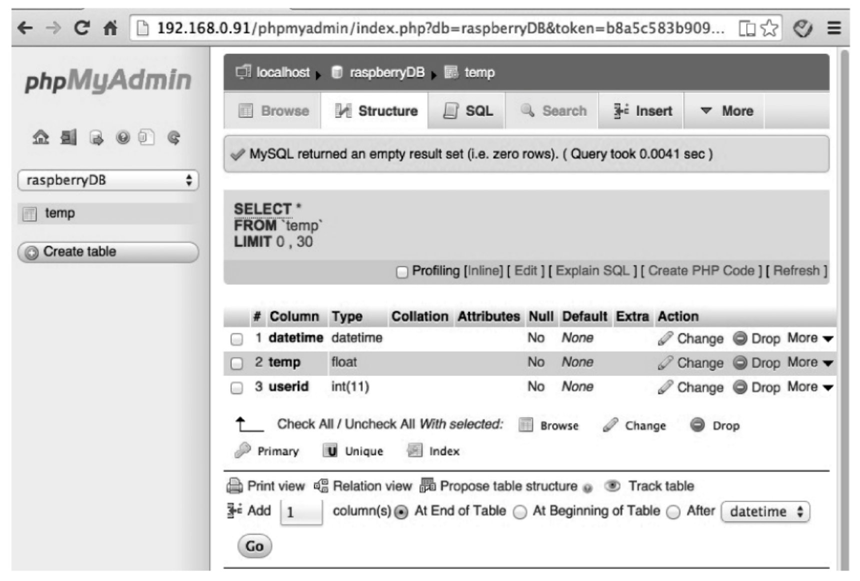


图 10-18　数据库表单的创建

程序解说

第 2 行：连接数据库，把“root”，“raspberry”修改为 MySQL 中的账号与密码。

第 3 行：获取现在的时间。

第 11～12 行：写入数据。

第 14 行：关闭数据库。

注意　第 2 行有指定数据库的设置，请按照实际情况自行调整。



```
GNU nano 2.2.6            File: settemp.php                  Modified

<?php
$con=mysqli_connect("localhost","root","raspberry","raspberryDB");
if (mysqli_connect_errno()) {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$now1= date('Ymdhis');
$id = $_GET['id'];
$temp = $_GET['temp'];
mysqli_query($con,"INSERT INTO temp (datetime,temp,userid)
  VALUES ($now1,$temp,$id)");

mysqli_close($con);
 echo "powenko.com get it".", date time=".$now.", temp=".$temp."  id=".$id;
?>
```



图 10-19　编写 PHP 程序

5. 结果

打开网页，输入以下网址进行测试，并把 IP 修改为实际的 IP 地址。如果成功，数据库就出现一条新的数据，参数 id 的意思是别名，temp 的意思是温度，如图 10-20 所示。

```
http://192.168.0.91/settemp.php?id = 1&temp = 31
```



图 10-20　测试

可以到数据库中查询是否新增了一条数据，如图 10-21 所示。

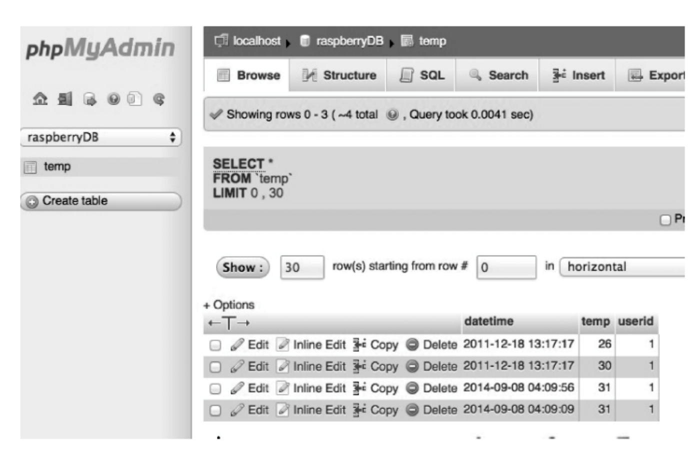


图 10-21　查询数据库进行确认

6. 教学影片

完整的教学视频可以看 RaspberryPi_Temperature_PHP_InsertDB。

10.6 物联网——通过网页显示数据——表格显示

案例 62

1. 目的

本案例介绍把放在 MySQL 数据库的数据,通过表格的方法显示出来。

2. 步骤

先成功执行案例 61,再继续以下的操作。

在网站服务器中,通过文本编辑器编写程序。

```
$ sudo su
  # cd /var/www
  # nano showtemp.php
```

输入以下程序,完成后,通过 Ctrl+O 组合键存储程序,Ctrl+X 组合键离开 nano 编辑器,如图 10-22 所示。

```
GNU nano 2.2.6                File: showtemp.php

<?php
$con=mysqli_connect("localhost","root","raspberry","raspberryDB");
if (mysqli_connect_errno()) {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
$result = mysqli_query($con,"SELECT * FROM temp");

echo "<table border='1'>
<tr>
<th>Date Time</th>
<th>Temperature</th>
<th>user ID </>
</tr>";
while($row = mysqli_fetch_array($result))  {
  echo "<tr>";
  echo "<td>" . $row['datetime'] . "</td>";
  echo "<td>" . $row['temp'] . "</td>";
  echo "<td>" . $row['userid'] . "</td>";
  echo "</tr>";
}
echo "</table>";
mysqli_close($con);
?>

                    [ Wrote 27 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 10-22 编写 PHP 程序

样例程序:sample\ch10\showtemp.php

```
1.  <?php
2.  $ con = mysqli_connect("localhost","root","raspberry","raspberryDB");
3.  if (mysqli_connect_errno()) {
4.    echo "Failed to connect to MySQL: " .  mysqli_connect_error();
5.  }
```

```
6.  $ result = mysqli_query( $ con,"SELECT * FROM temp");
7.
8.  echo "< table border = '1'>
9.  < tr >
10. < th > Date Time </th >
11. < th > Temperature </th >
12. < th > user ID </>
13. </tr >";
14. while( $ row = mysqli_fetch_array( $ result))  {
15. echo "< tr >";
16. echo "< td >" .  $ row[ 'datetime'] .  "</td >";
17. echo "< td >" .  $ row[ 'temp'] .  "</td >";
18. echo "< td >" .  $ row[ 'userid'] .  "</td >";
19. echo "</tr >";
20. }
21. echo "</table >";
22. mysqli_close( $ con);
23. ?>
24.
```

程序解说

第 2 行：连接数据库，把"root"，"raspberry"修改为 MySQL 中的账号与密码。

第 6 行：获取数据库所有的数据。

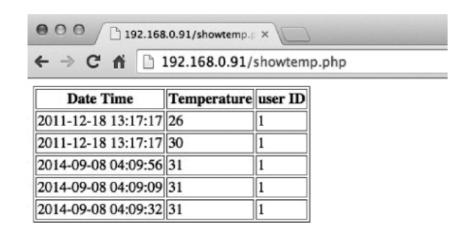
第 14～20 行：把数据显示在表格中。

第 22 行：关闭数据库。

3. 结果

打开网页，输入以下网址进行测试，并把 IP 修改为实际的 IP 地址。如果成功，就会显示数据库中的所有数据，如图 10-23 所示。

```
http://192.168.0.91/showtemp.php
```



Date Time	Temperature	user ID
2011-12-18 13:17:17	26	1
2011-12-18 13:17:17	30	1
2014-09-08 04:09:56	31	1
2014-09-08 04:09:09	31	1
2014-09-08 04:09:32	31	1

图 10-23 测试

4. 教学影片

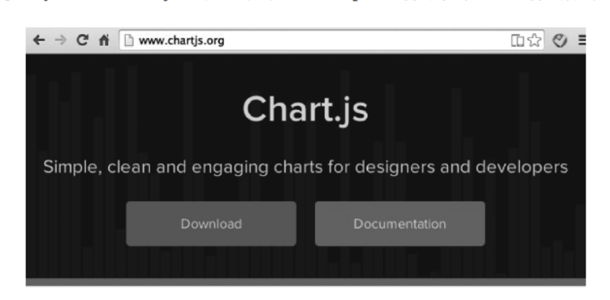
完整的教学视频可以看 RaspberryPi_showDB1。

10.7 物联网——通过网页显示数据——图形显示

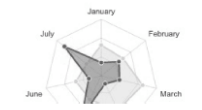
案例 63

1. 目的

本案例延续显示数据库的操作,介绍把放在 MySQL 数据库的数据,通过图形化的方法显示出来。使用 jQuery 和 Chart.js 这两个 JavaScript 函数库来显示数据,如图 10-24 所示。



Six chart types



HTML5 based



图 10-24 官方网站 http://www.chartjs.org/

2. 步骤

先成功执行案例 62,再继续以下的操作。

STEP1:下载 jQuery 图形函数库 Chart.js

进入网站服务器的/var/www 中,下载 Chart.js 的原始程序。

```
$ sudo su
# cd /var/www
# wget https://github.com/nnnick/Chart.js/archive/master.zip
```

STEP2:解压缩

解压缩执行如下程序。

```
# unzip master.zip
```

STEP3:编写 PHP 程序

切换到网站服务器的网页路径中,通过文本编辑器编写程序。

```
pi@raspberrypi /var/www $ sudo su
root@raspberrypi:/var/www# cd /var/www
root@raspberrypi:/var/www# wget https://github.com/nnnick/Chart.js/archive/maste
r.zip
--2014-09-08 17:27:29--  https://github.com/nnnick/Chart.js/archive/master.zip
Resolving github.com (github.com)... 192.30.252.129
Connecting to github.com (github.com)|192.30.252.129|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/nnnick/Chart.js/zip/master [following]
--2014-09-08 17:27:33--  https://codeload.github.com/nnnick/Chart.js/zip/master
Resolving codeload.github.com (codeload.github.com)... 192.30.252.145
Connecting to codeload.github.com (codeload.github.com)|192.30.252.145|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: 92450 (90K) [application/zip]
```

图 10-25　下载 jQuery 图形函数库 Chart. js

```
$ sudo su
# cd /var/www
# nano showtempChart.php
```

输入以下程序,完成后,通过 Ctrl+O 组合键存储程序,Ctrl+X 组合键离开 nano 编辑器,如图 10-26 所示。

```
GNU nano 2.2.6            File: showtempChart.php

<!doctype html>
<html>
        <head>
                <title>Bar Chart</title>
                <script src="Chart.js-master/Chart.js"></script>
        </head>
        <body>
                <div style="width: 50%">
                        <canvas id="canvas" height="450" width="800"></canvas>
                </div>
<?php
$con=mysqli_connect("localhost","root","raspberry","raspberryDB");
if (mysqli_connect_errno()) {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
}


$result = mysqli_query($con,"SELECT * FROM temp");

echo "<table border='1'>
<tr>
<th>Date Time</th>
<th>Temperature</th>
<th>user ID </>
                               [ Read 74 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 10-26　编写 PHP 程序

样例程序:sample\ch10\showtempChart. php

```
1.  <!doctype html>
2.  <html>
3.          <head>
4.                  <title>Bar Chart</title>
```

```
5.                  <script src="Chart.js-master/Chart.js"></script>
6.          </head>
7.          <body>
8.                  <div style="width: 50%">
9.                          <canvas id="canvas" height="450" width="800"></canvas>
10.                 </div>
11. <?php
12.  $con = mysqli_connect("localhost","root","raspberry","raspberryDB");
13. if (mysqli_connect_errno()) {
14.   echo "Failed to connect to MySQL: " . mysqli_connect_error();
15. }
16.
17.
18.
19.  $result = mysqli_query($con,"SELECT * FROM temp");
20.
21. echo "<table border='1'>
22. <tr>
23. <th>Date Time</th>
24. <th>Temperature</th>
25. <th>user ID </>
26.
27. </tr>";
28.
29. while($row = mysqli_fetch_array($result))
30. {
31.  echo "<tr>";
32.  echo "<td>" . $row['datetime'] . "</td>";
33.  echo "<td>" . $row['temp'] . "</td>";
34.  echo "<td>" . $row['userid'] . "</td>";
35.  echo "</tr>";
36.   $Lables = $Lables.'"'. $row['datetime'].'",';
37.   $temps = $temps.'"'. $row['temp'].'",';
38. }
39.
40. echo "</table>";
41. mysqli_close($con);
42. ?>
43.
44.
45.         <script>
46. var barChartData = {
47.                 labels : [<?php echo  $Lables;  ?>],
48.                 datasets : [
49.                         {
50.                                 fillColor : "rgba(20,20,20,0.5)",
```

```
51.                                     strokeColor : "rgba(220,220,220,0.8)",
52.                                     highlightFill: "rgba(220,220,220,0.75)",
53.                  highlightStroke: "rgba(220,220,220,1)",
54.                                     data : [<?php echo  $ temps;  ?>
55.                                     ]
56.
57.                             }
58.                     ]
59.
60.         }
61.         window.onload = function(){
62.                 var ctx = document.getElementById("canvas").getContext("2d");
63.                 window.myBar = new Chart(ctx).Bar(barChartData,{
64.                         responsive : true
65.                 });
66.         }
67.         </script>
68.         </body>
69.  </html>
70.
```

程序解说

第 12 行：连接数据库，把“root”，“raspberry”修改为 MySQL 中的账号与密码。

第 19 行：获取数据库所有的数据。

第 29～38 行：把数据显示在表格中。

第 36～37 行：把数据相加。

第 41 行：关闭数据库。

第 46～60 行：转换成 JSON 的数据。

第 63 行：显示图形。

3. 结果

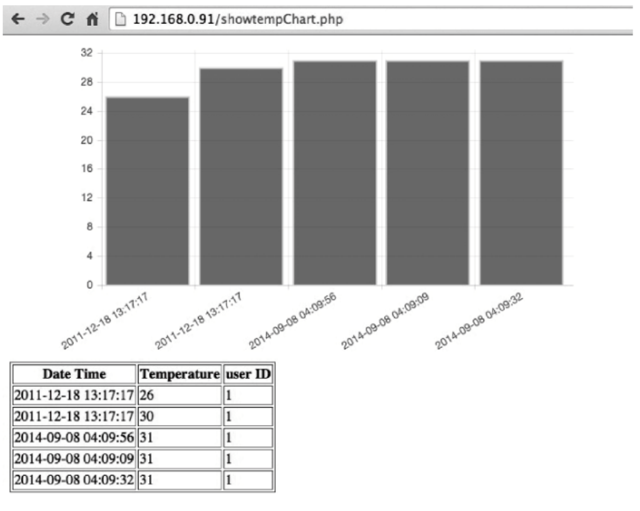
打开网页，输入以下网址进行测试，并把 IP 修改为实际的 IP 地址。如果成功，就会显示数据库中的所有数据和图形，如图 10-27 所示。

```
http://192.168.0.91/showtempChart.php
```

4. 教学影片

完整的教学视频可以看 RaspberryPi_showDB2_chart。

注意　jQuery 和 Chart.js 这两个 JavaScript 函数库拥有各种图表显示样式，各位可以按照实际的需要调整程序，使表达出来的结果更容易让用户理解。



Date Time	Temperature	user ID
2011-12-18 13:17:17	26	1
2011-12-18 13:17:17	30	1
2014-09-08 04:09:56	31	1
2014-09-08 04:09:09	31	1
2014-09-08 04:09:32	31	1

图 10-27　测试结果

10.8　物联网——温度监测系统

案例 64

1. 概述

物联网的温度监测系统几乎要完成了，此案例就把所有的东西放在一起给温度监测系统做个整合来具体完成。

温度检测的设备可以放在远程，只要通过网络把数据返回给主机就可以了，这样的技术可以运用在很多地方，例如

(1) 电冰箱内的温度检测，确保冰箱的温度够低，不利于细菌繁殖。

(2) 农田大棚的温室检测，确保高级农产品在温室内的气温，在一定的安全范围内。

(3) 计算机机房的温度，确保机器散热正常。

(4) 任何需要记录温度的地方等。

2. 准备

为了完成本案例需要先了解：

(1) 设置固定网络 IP，请查看“3.1 设置固定 IP 网络”。

(2) 搭建网站服务器，请查看“4.1 搭建 PHP 网站服务器”中的 PHP 网站搭建。

(3) 安装 MySQL 数据库，请查看“4.3 创建数据库 MySQL ”中的 MySQL 安装。

(4) 温度计的硬件，请查看“10.2 使用温度传感器监测室内温度”中的相关硬件安装。

3. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 1 个面包板;

(3) 接线若干;

(4) 网络设备。

4. 步骤

首先接好硬件线路。

STEP1:确认写入数据的功能

首先参考"10.5 物联网——把数据写到数据库"通过网页确认树莓派写入数据库的功能正常,如图 10-28 所示。

```
http://192.168.0.91/settemp.php?id = 1&temp = 31
```



图 10-28　输入树莓派写入数据库的网址

STEP2:确认数据上传到网络的功能

修改"10.4 物联网——把数据上传到网络"的网址部分,通过文本编辑器打开 temp_url.py 程序,如图 10-29 所示。

```
$ nano temp_url.py
```

修改 IP 和网页地址。

```
content,response_code = fetch_thing(
                          'http://192.168.0.91/settemp.php/',
                          {'id': 1,'temp': temp},
                          'GET'
                      )
```

```
a6 = GPIO.input(18)
a7 = GPIO.input(22)
total=a0+(a1*2)+(a2*4)+(a3*8)+(a4*16)+(a5*32)+(a6*64)+(a7*128)
# temp=total*(5*1000/256)/10;
temp=total*5*1000/256/10;
print a7,a6,a5,a4,a3,a2,a1,a0,"[",total,"]","[C=",temp,"]"
#  time.sleep(0.1)
content, response_code = fetch_thing(
                              'http://192.168.0.91/settemp.php/',
                              {'id': 1, 'temp': temp},
                              'GET'
                         )

# http://192.168.0.91/settemp.php?id=1&temp=31
```

图 10-29　修改 temp_url.py 程序 IP 和网页地址

STEP3:确认上传时间

修改"10.3 定时执行程序"调用程序的时间。输入以下命令设置 Shell 的执行时间,如图 10-30 所示。

```
$ sudo crontab -e
```

树莓派就会自动打开 nano 文本编辑器,在文本最后添加以下命令,如图 10-30 所示。

```
*/1 * * * * /home/pi/code/job.sh
```

意思是每分钟执行/home/pi/code/job.sh 的 Shell Script 一次。完成后,通过 Ctrl+O 组合键存储,Ctrl+X 组合键离开编辑器。

```
GNU nano 2.2.6        File: /tmp/crontab.KJX4cs/crontab

# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
# 1.M   2.H    3.D    4.M    5.W
#  M = minute  (0 - 59)2. S = hour (0 - 23)3. D = day
#    (1 - 31)4. M = month
#    (1 - 12)5. W=weekday (0 - 7) (Sonntag is represented by 0 and 7)
# m h  dom mon dow   command

*/1 * * * *  /home/pi/code/job.sh

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 10-30 确认上传时间

修改成每小时执行一次

```
* */1 * * * /home/pi/code/job.sh
```

修改成每天 12 点执行一次

```
0 12 * * * /home/pi/code/job.sh
```

STEP4:确认定时工作

通过文本编辑器确认定时工作。

```
$ sudo nano ./job.sh
```

设置程序去执行温度计数据上传的 Python 程序,如图 10-31 所示。

```
sudo python /home/pi/code/temp_url.py
```

执行以下命令设置文件权限。

```
$ sudo chmod 755 job.sh
```

```
 GNU nano 2.2.6          File: ./job.sh                    Modified

sudo  python /home/pi/code/temp_url.py





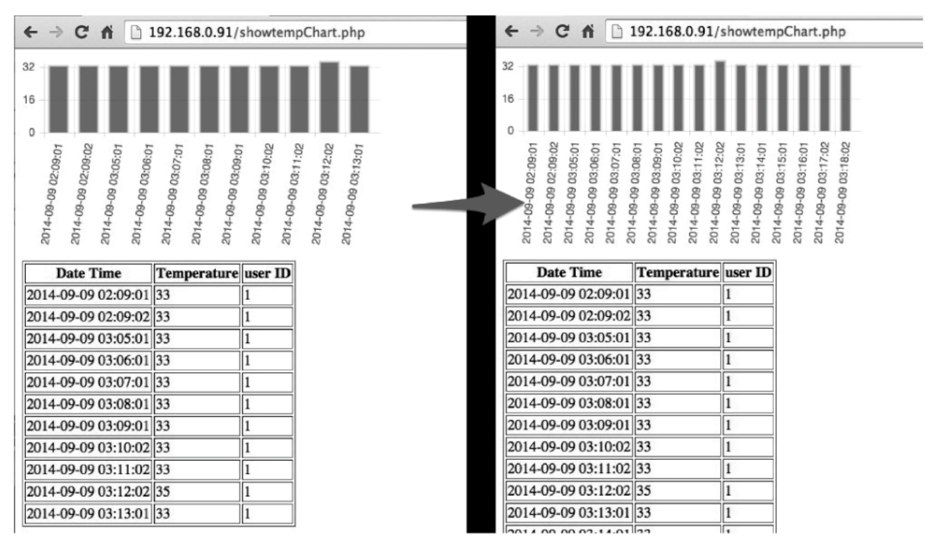
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 10-31　code/job.sh 的内容

5. 结果

打开网页,输入以下网址进行测试,并把 IP 修改为实际的 IP 地址。如果成功,就会显示数据库中的所有数据和图形,如图 10-32 所示。

```
http://192.168.0.91/showtempChart.php
```



Date Time	Temperature	user ID
2014-09-09 02:09:01	33	1
2014-09-09 02:09:02	33	1
2014-09-09 03:05:01	33	1
2014-09-09 03:06:01	33	1
2014-09-09 03:07:01	33	1
2014-09-09 03:08:01	33	1
2014-09-09 03:09:01	33	1
2014-09-09 03:10:02	33	1
2014-09-09 03:11:02	33	1
2014-09-09 03:12:02	35	1
2014-09-09 03:13:01	33	1

Date Time	Temperature	user ID
2014-09-09 02:09:01	33	1
2014-09-09 02:09:02	33	1
2014-09-09 03:05:01	33	1
2014-09-09 03:06:01	33	1
2014-09-09 03:07:01	33	1
2014-09-09 03:08:01	33	1
2014-09-09 03:09:01	33	1
2014-09-09 03:10:02	33	1
2014-09-09 03:11:02	33	1
2014-09-09 03:12:02	35	1
2014-09-09 03:13:01	33	1

图 10-32　测试

也可以通过智能手机或平板电脑来显示温度的变化,如图 10-33 所示。

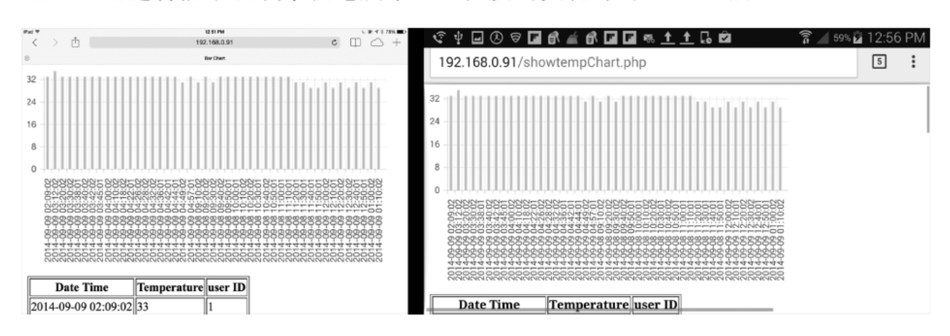


图 10-33　分别在 iOS 和 Android 上的效果

6. 教学影片

完整的教学视频可以看 RaspberryPi_Temp_Final。

注意　用网络来存储和显示数据,最重要是符合物联网的精神,把数据通过网络传递并可以多平台汇整在一起。如果只有一台,可以用 IP 127.0.0.1 把数据传递给自身进行处理。

第11章 树莓派照相机与摄像机——内置Camera应用

11.1 树莓派照相机

案例65

1. 介绍

在柯老师第一本《Raspberry Pi最佳入门与实战应用》的书中,有提到使用USB的Webcam来拍照,本案例会用树莓派板子上的内置Camera来拍照。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 一个内置Camera。

需要有以下软件:

```
ownCloud
```

树莓派的摄像头模块是500万像素,并针对树莓派的照相机所设计,外观如图11-1所示。

它采用了定焦镜头,能够拍摄2592×1944像素的静态图片,同时还支持1080p+30Fps、720p+60Fps及640×480p+60/90Fps的画面尺寸和速度Fps(Frame per second)的视频。特别使用板子表面的专用CSI接口来连接照相机。

这个板子本身很小,大约为25mm×20mm×9mm,非常适合移动应用,通过一个短的带状电缆连接到树莓派上面。

传感器本身具有的500万像素,并板载一个定焦镜头,能够拍摄2592×1944像素的静态图片。

摄像机使用OmniBSI技术,高性能1.4μm×1.4μm像素,1/4光学尺寸,自动图像控制功能:

(1) 自动曝光控制(AEC);

(2) 自动白平衡(AWB);

(3) 自动带式过滤器(ABF);

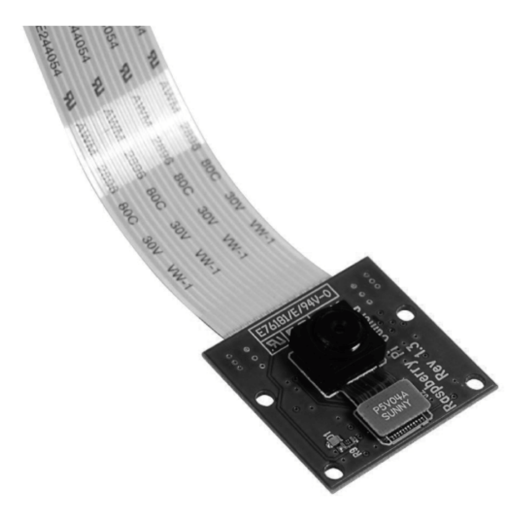


图 11-1　树莓派 ver 1.3 的 500 万像素摄像机

(4) 自动 50/60Hz Luminance 检测;

(5) 自动黑电平校正(ABLC)。

3. 接线

把内接式照相机与树莓派开发板上的 CSI 接口相连。

4. 步骤

STEP1:硬件接线

首先关闭电源,再通过树莓派板子上的 CSI 接口和摄像机连接。摄像机 CSI 排线一边有电线,一边没有,连接靠近网线接口旁的 CSI 接口,有电线的那一面接在对面板子即 HDMI 那边,也就是蓝色那一边靠近网络孔,如图 11-2 所示。

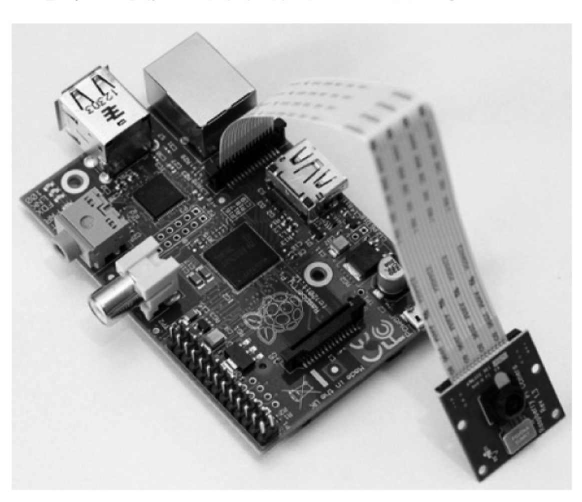


图 11-2　摄像机 CSI 排线连接

STEP2：打开 Camera 的设置

首先进入设置页。

```
$ sudo raspi - config
```

选择 Enable Camera（引导照相机），如图 11-3 所示。

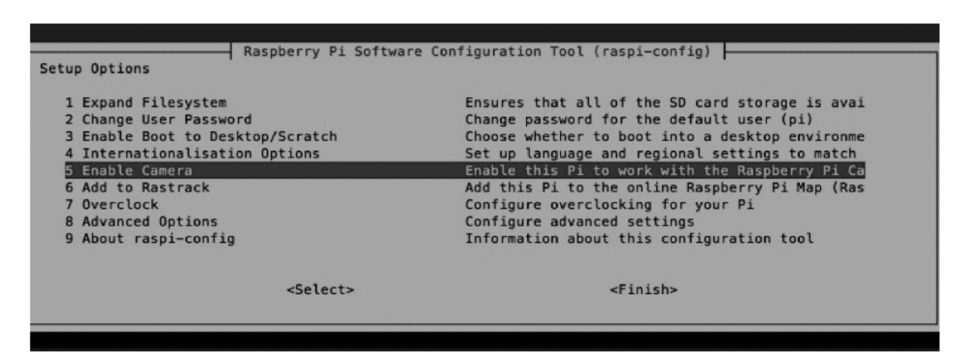


图 11-3　选择 Enable Camera

然后通过 Tab 按键，选择 Enable 选项，按下 Enter 按键，如图 11-4 所示。

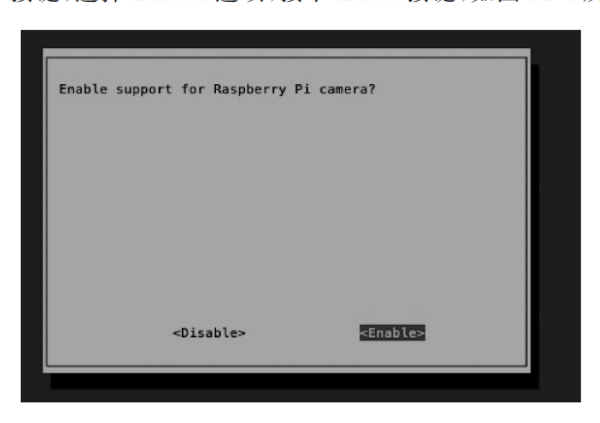


图 11-4　选择 Enable

回到设置主菜单后，再选择 Finish 选项，重新引导树莓派。

STEP3：获取 Camera 的影像

重新开机后，可以输入以下命令获取图片。

```
$ raspistill - o image.jpg
```

5. 结果

输入以下命令获取图片 image.jpg，如果要更换文件名，把 image.jpg 换成其他名称即可，如图 11-5 所示。

```
$ raspistill －o image.jpg
```

```
pi@raspberrypi ~ $ rm image.jpg
pi@raspberrypi ~ $ ls
c  code  Desktop  download  linux  ocr_pi.png  python_games  samplecode  screen.png  src  tool  tools
pi@raspberrypi ~ $ raspistill -o image.jpg
pi@raspberrypi ~ $ ls
c     Desktop   image.jpg  ocr_pi.png    samplecode  src   tools
code  download  linux      python_games  screen.png  tool
```

图 11-5　执行结果

在获取文件时,屏幕会有 5 秒钟显示 Camera 的画面,如图 11-6 所示。



图 11-6　执行画面

6. 教学影片

完整的教学视频可以看 RaspberryPi_05Camera.mov,实际硬件情况,如图 11-7 所示。

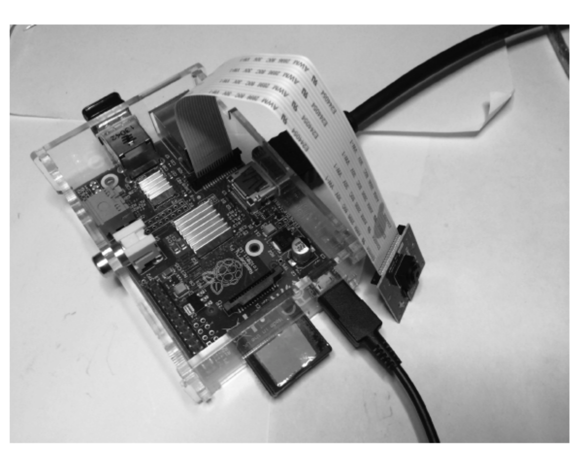


图 11-7　实际硬件情况

11.2 树莓派摄像机

案例 66

1. 介绍

本案例将会用树莓派板子上的内置 Camera 接口来录制影片,并且设置影片尺寸和录制的时间长度。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络连接设备;

(3) 内接式的照相机。

安装以下软件:

Raspivid

3. 准备

本案例执行前需要先完成"11.1 树莓派照相机"。

4. 接线

把外接硬盘接到 USB 接口上。

5. 步骤

通过以下样例的命令获取影片。录制后的影片为 H.264 格式,可以通过 VLC 等播放器观看影片内容。Raspivid 软件是系统自带软件,不需要另外安装。

1) 样例

录制 5 秒钟的影片,并存在 1.h264 文件。用内置 Camera 照相机,因为程序默认的录制时间是 5 秒钟,所以影片长度为 5 秒。

```
$ raspivid -o 1.h264
```

2) 样例

录制 10 秒钟的影片,并存在 2.h264 文件。用内置 Camera 照相机,录制 10 秒钟的影片。

```
$ raspivid -o 2.h264 -t 10000s
```

3) 样例

录制 10 秒钟的影片,并存在 3.h264 文件,并指定影片宽度为 320 高度为 240。用内置 Camera 照相机,录制 10 秒钟的影片,如图 11-8 所示。

```
$ raspivid -o 3.h264 -w 320 -h 240 -t 10000s
```

```
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264 -t 10s
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264 -t 10s
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264 -t 10
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264 -t 10s
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264 -t 1000s
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264 -t 10000s
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264 -t 1000
mmal: Unable to return a buffer to the encoder port
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264 -t 1000s
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 2.h264 -t 10000s
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ raspivid -o 3.h264 -w 320 -h
240 -t 10000s
pi@raspberrypi ~/samplecode/RaspberryPi/10Record $ ls
1.h264  2.h264  3.h264
```

图 11-8 执行命令

6. 结果

输入命令后就可以录制影片,并且录制时会在屏幕上显示摄像的内容,如图 11-9 所示。

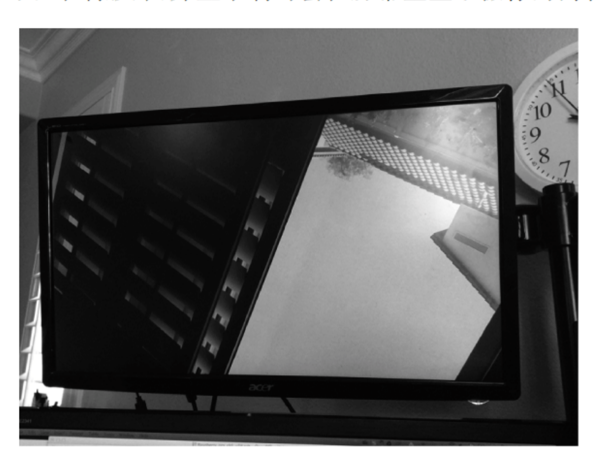


图 11-9 执行画面

7. 教学影片

完整的教学视频可以看 raspberryPi_raspivid.mov。

11.3 编写 Camera 拍照和录像的 Python 程序

案例 67

1. 目的

本案例介绍使用 Python 程序语言,并使用 picamera 的模块,编写程序调整一些参数和功能。

2. 步骤

继续案例 66 和硬件接线。为了可以使用 picamera 的模块,需要先安装一些相关的模块。

```
$ sudo rpi - update
```

STEP1：下载安装 ez_setup 的程序

```
$ wget https://bootstrap.pypa.io/ez_setup.py -O - | python
$ unzip setuptools - 3.5.1.zip
$ cd setuptools - 3.5.1/
$ sudo python ./setup.py install
```

STEP2：下载安装 pytest 模块

输入以下命令下载安装 pytest 模块，如图 11-10 所示。

```
$ easy_install -U pytest
```

```
pi@raspberrypi ~/samplecode/RaspberryPi/9picamera/picamera-master/tests $ sudo easy_install -U pytest
Searching for pytest
Reading https://pypi.python.org/simple/pytest/
Best match: pytest 2.5.2
Downloading https://pypi.python.org/packages/source/p/pytest/pytest-2.5.2.tar.gz#md5=8ea3d1939e81514ccba9ba0e9566b5be
Processing pytest-2.5.2.tar.gz
Writing /tmp/easy_install-XMjLwD/pytest-2.5.2/setup.cfg
Running pytest-2.5.2/setup.py -q bdist_egg --dist-dir /tmp/easy_install-XMjLwD/pytest-2.5.2/egg-dist-tmp-hYUEWI
Adding pytest 2.5.2 to easy-install.pth file
Installing py.test-2.7 script to /usr/local/bin
Installing py.test script to /usr/local/bin
```

图 11-10　下载安装 pytest

STEP3：下载安装 Python Image 模块

```
$ sudo apt - get install python - imaging
```

STEP4：下载安装 GPIO 等函数库

```
$ sudo apt - get install python - picamera python3 - picamera python - rpi.gpio
```

STEP5：下载 picamera 的程序

安装新的程序和模块 picamera，先创建一个空白路径，通过 wget 下载 Python 的程序和模块 picamera，如图 11-11 所示。

```
$ wget https://github.com/waveform80/picamera/archive/master.zip
```

```
pi@raspberrypi ~/samplecode/RaspberryPi/9picamera $ wget https://github.com/wave
form80/picamera.git
--2014-05-06 21:52:58--  https://github.com/waveform80/picamera.git
Resolving github.com (github.com)... 192.30.252.130
Connecting to github.com (github.com)|192.30.252.130|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/waveform80/picamera [following]
--2014-05-06 21:53:02--  https://github.com/waveform80/picamera
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 200 OK
Length: 49572 (48K) [text/html]
Saving to: `picamera.git'

100%[======================================>] 49,572      114K/s   in 0.4s

2014-05-06 21:53:06 (114 KB/s) - `picamera.git' saved [49572/49572]
```

图 11-11　下载 picamera

注意 picamera 的原始程序可以在这里下载 https://github.com/waveform80/picamera。

STEP6：解压缩

输入以下命令解开压缩文件，如图 11-12 所示。

```
$ unzip master.zip -d picamera
```

```
pi@raspberrypi ~/samplecode/RaspberryPi/9picamera $ ls
master.zip  readme.txt
pi@raspberrypi ~/samplecode/RaspberryPi/9picamera $ unzip master.zip -d picamera
Archive:  master.zip
b8010793794aeca85f7754a2c017a2d4c2f93b7f
   creating: picamera/picamera-master/
  inflating: picamera/picamera-master/.gitignore
  inflating: picamera/picamera-master/LICENSE.txt
  inflating: picamera/picamera-master/MANIFEST.in
```

图 11-12 解压缩

STEP7：安装 picamera

输入以下命令安装 picamera。

```
$ cd picamera
$ sudo python ./setup.py install
```

STEP8：编写拍照的 pythin 程序

样例程序：sample\ch9\01image.py

```
1.  import time
2.  import picamera
3.
4.  with picamera.PiCamera() as camera:
5.      camera.start_preview()
6.      time.sleep(5)
7.      camera.capture('image.jpg')
8.      camera.stop_preview()
```

程序解说

第 5 行：预览。

第 6 行：停止 5 秒钟。

第 7 行：抓取图片到 image.jpg。

第 8 行：关闭预览，离开程序。

输入以下命令抓取 Camera 上的照片，如图 11-13 所示。

```
$ sudo python ./01image.py
```

```
pi@raspberrypi ~/samplecode/RaspberryPi/9picamera/image $ sudo python 01image.py
pi@raspberrypi ~/samplecode/RaspberryPi/9picamera/image $ ls
01image.py  image.jpg
```

图 11-13 01image. py 执行情况

STEP9：编写录像的 pythin 程序

样例程序：sample\ch9\02video. py

```
1.  # www.powenko.com
2.  import time
3.  import picamera
4.  import RPi.GPIO as GPIO
5.
6.  #GPIO.setmode(GPIO.BCM)
7.  #GPIO.setup(17,GPIO.IN,GPIO.PUD_UP)
8.
9.  with picamera.PiCamera() as camera:
10.     camera.start_preview()
11.     #GPIO.wait_for_edge(17,GPIO.FALLING)
12.     camera.resolution = (640,480)
13.     camera.start_recording('my_video.h264',inline_headers = False)
14.     #GPIO.wait_for_edge(17,GPIO.FALLING)
15.     camera.wait_recording(60)
16.     camera.stop_recording()
17.     camera.stop_preview()
```

程序解说

第 10 行：预览。

第 12 行：设置影片尺寸。

第 13 行：录像，并指定影片名称。

第 15 行：录制 60 秒。

第 16～17 行：关闭预览，离开程序。

这个程序有几行注解掉的代码，是用 GPIO 17 的引脚做按键录像的操作，如果想改成可通过硬件按键来录像，就可以把注解拿掉。

输入以下命令就可以录像了。

```
$ sudo python ./02video.py
```

3. 结果

执行结果如图 11-14 所示。



图 11-14　执行结果

11.4　相机外壳设计

11.4.1　纸相机

案例 68

这段时间看到很多学生的新文化创意作品，每个都很精致和充满创意，所以就在想是否有可能将树莓派与文创商品做个结合，让各位在本书所学习的知识和文创结合产生更高的价值。所以就产生用树莓派拍照的功能加上厚纸板的设计外壳做出很有趣的一些相机造型的想法。

本案例需要先完成案例 67，接下来就是把这些设备放在外壳之中，制作外壳的方法，例如在本书的范例中找到 paperCamera.pdf 这个文件，使用打印机打印出来贴在厚纸板上，剪切下来就是一个照相机的外壳，如图 11-15 所示。

样例文件：sample\ch10\paperCamera.pdf

11.4.2　3D 打印机

案例 69

如果手上有 3D 打印机，也可以下载 3D 模型文件 STL，调整一下就能成为树莓派照相机的外壳，模型文件下载地址为 http://www.thingiverse.com/thing:244361/#files，如图 11-16 所示。

下载后，放入 3D 打印机的软件中，调整输出位置，确认后按下输出，如图 11-17 所示。

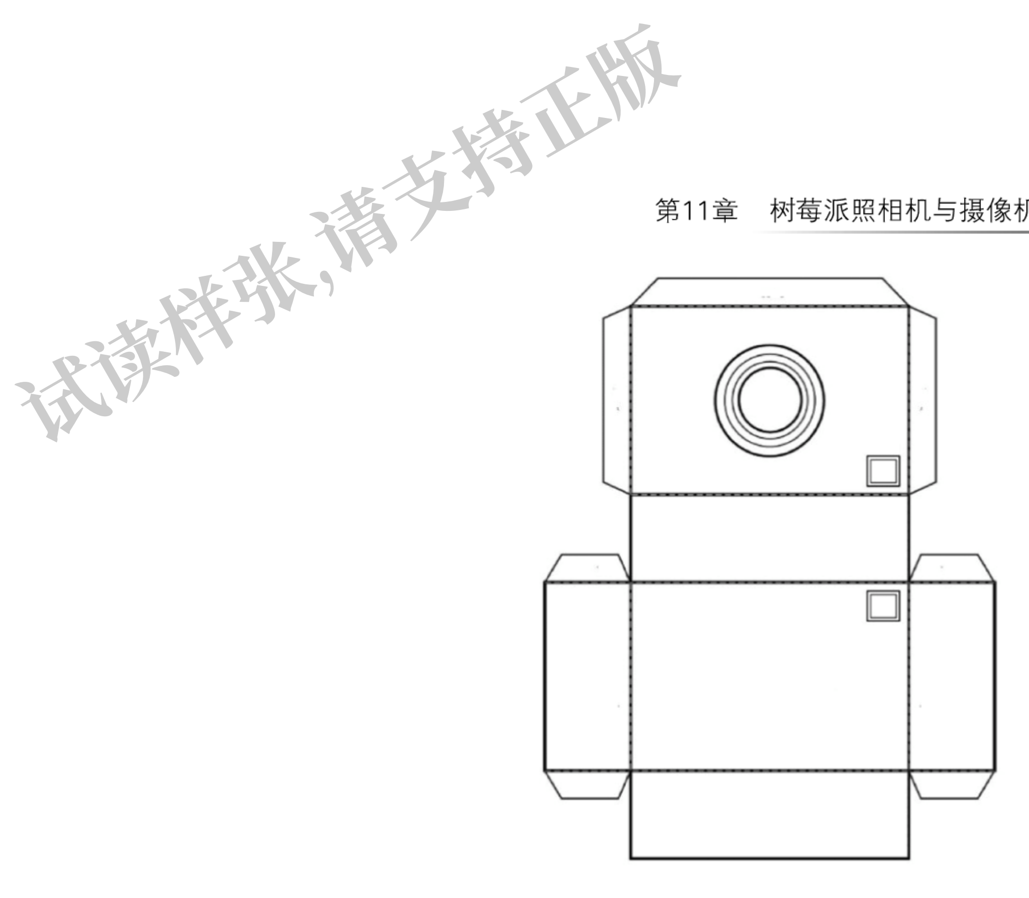


图 11-15　文件 paperCamera. pdf 厚纸板外壳

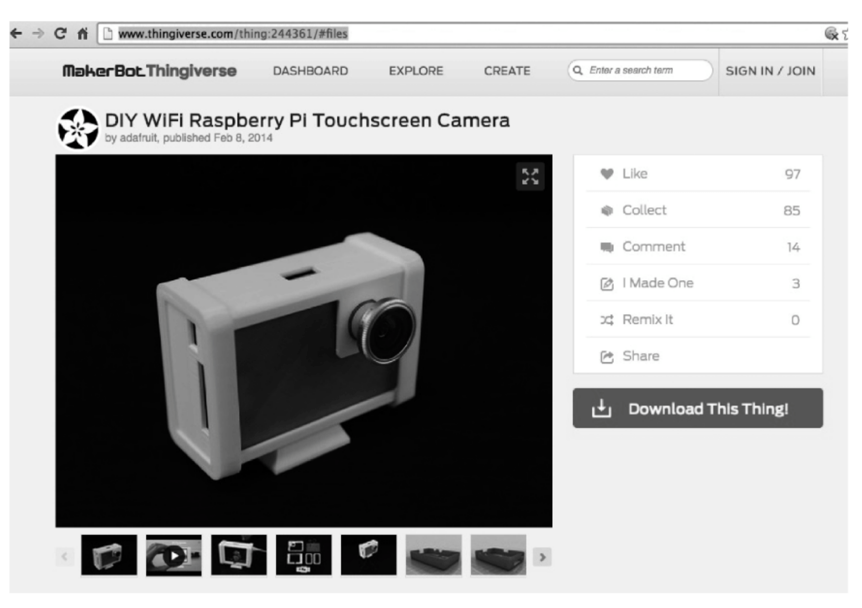


图 11-16　树莓派照相机的外壳模型下载

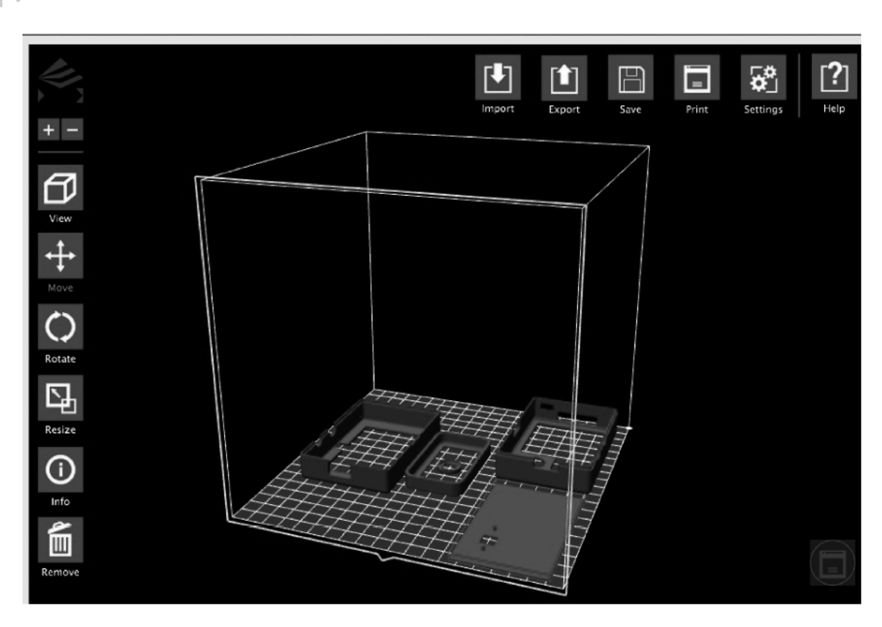


图 11-17　3D 打印机软件中的照相机外壳

接下来大约等待 3 个小时成品就完成了，如图 11-18 所示。

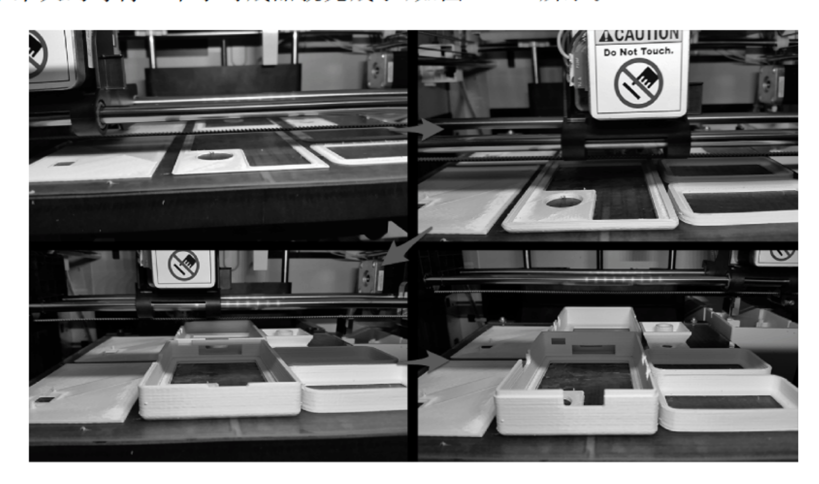


图 11-18　制作过程

把外壳、摄像机镜头与树莓派组装完毕后，就会是一台很棒的树莓派照相机，如图 11-19 所示。



图 11-19　组装和成品

11.5　摄像机头外壳

案例 70

如果已经有现成的树莓派的外壳,只设计树莓派的摄像机头的外壳,可以在 http://www.thingiverse.com/thing:92208 下载 STL 模型文件,如图 11-20 所示。

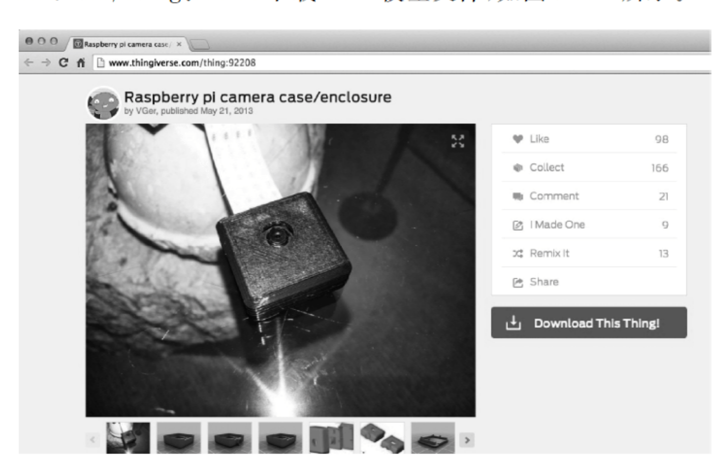


图 11-20　Vger 所设计的树莓派的摄像机头的外壳

下载 STL 文件后,放入 3D 打印机中输出,大约等待 20 分钟,如图 11-21 所示。

打印出成品如图 11-22 所示,成品的毛边稍微修整一下,用美工刀把不必要的部分切除。

完成之后的保护外壳如图 11-23 所示。



图 11-21　正在通过 3D 打印机打印出成品



图 11-22　3D 打印机打印出的成品



图 11-23　完成

第 12 章

多媒体播放器

12.1 调整音量

案例 71

1. 介绍

树莓派调整声音大小没有硬件的旋钮,需要用软件来调整音量大小配合声音输出的设备,本案例介绍如何处理和设置声音。

2. 步骤

STEP1:升级安装程序

安装 VLC 之前,更新系统的安装程序 apt-get。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

STEP2:安装 PiAUISuite 的声音播放工具

PiAUISuite 是个开源代码,主要是提供树莓派的声音功能,需要安装它来控制声音的大小,如图 12-1 所示。如果 git 的路径已被修改,到官方网站 https://github.com/StevenHickson/PiAUISuite 查找新的 git。

```
$ sudo apt - get install git - core
$ git clone git://github.com/StevenHickson/PiAUISuite.git
$ cd PiAUISuite/Install/
$ ./InstallAUISuite.sh
```

安装的 InstallAUISuite 会询问一堆问题,只要按照问题小心回答就可以了。

STEP3:打开声卡调整音量

输入以下的命令打开声卡。

```
$ modprobe snd_bcm2835
```

调整声音的设备。

```
pi@raspberrypi ~ $ git clone git://github.com/StevenHickson/PiAUISuite.git
Cloning into 'PiAUISuite'...
remote: Reusing existing pack: 657, done.
remote: Total 657 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (657/657), 3.51 MiB | 753 KiB/s, done.
Resolving deltas: 100% (359/359), done.
pi@raspberrypi ~ $ sudo git clone git://github.com/StevenHickson/PiAUISuite.git
fatal: destination path 'PiAUISuite' already exists and is not an empty directory.
pi@raspberrypi ~ $ ls
code  Desktop  indiecity  ocr_pi.png  PiAUISuite  python_games
pi@raspberrypi ~ $ cd PiAUISuite/
pi@raspberrypi ~/PiAUISuite $ ls
DownloadController  Install  Makefile  PlayVideoScripts  TextCommand   Youtube
Imaging             LICENSE  Misc      README            VoiceCommand
pi@raspberrypi ~/PiAUISuite $ cd Install/
pi@raspberrypi ~/PiAUISuite/Install $ ls
InstallAUISuite.sh  UninstallAUISuite.sh  UpdateAUISuite.sh
pi@raspberrypi ~/PiAUISuite/Install $ ./InstallAUISuite.sh
Installing AUI Suite by Steven Hickson
If you have issues, visit stevenhickson.blogspot.com or email help@stevenhickson.com
Install dependencies? y/n
These are necessary for any of the options, so you should probably press y unless you absolutely know you have them already
y
Reading package lists... Done
```

图 12-1 下载和安装 PiAUISuite 软件和代码

```
$ amixer cset numid = 3 1
```

这里的 1 是输出声音的设备：

0 如果有接 HDMI，就是 HDMI 输出，否则就是耳机输出。

1 耳机接口。

2 HDMI 输出。

调整音量。

```
$ sudo amixer cset numid = 1 60 %
```

声音的大小范围在 0%～100%之间。

12.2 多媒体播放器——MPlayer

案例 72

1. 介绍

本案例将会介绍鼎鼎大名的多媒体软件 MPlayer，它能播放几乎所有流行的音频和视频格式。

2. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 网络连接设备；

(3) 一般的喇叭或者耳机。

安装以下软件：

(1) Raspivid；

(2) VLC。

3. 接线

确认树莓派的网络连接功能正常。

4. 步骤

STEP1:升级安装程序

在安装 VLC 之前,更新系统的安装程序 apt-get。

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

STEP2:安装 VLC

输入以下的命令,在树莓派上安装 VLC。

```
$ sudo apt-get install vlc
```

STEP3:安装 MPlayer

输入以下的命令,在树莓派上安装 MPlayer 并重新开机,如图 12-2 所示。

```
$ sudo apt-get install mplayer mplayer-gui alsa-base alsa-utils pulseaudio mpg123
$ sudo reboot
```

```
pi@raspberrypi ~ $ sudo apt-get install mplayer mplayer-gui alsa-base alsa-utils
 pulseaudio mpg123
```

图 12-2 安装 MPlayer

STEP4:下载测试影片

输入以下命令,下载测试用的影片,如图 12-3 所示。

```
$ wget http://mirrorblender.top-ix.org/peach/bigbuckbunny_movies/big_buck_bunny_480p_
h264.mov
```

```
pi@raspberrypi ~ $ wget http://mirrorblender.top-ix.org/peach/bigbuckbunny_movie
s/big_buck_bunny_480p_h264.mov
--2014-09-14 19:29:13--  http://mirrorblender.top-ix.org/peach/bigbuckbunny_movi
es/big_buck_bunny_480p_h264.mov
Resolving mirrorblender.top-ix.org (mirrorblender.top-ix.org)... 194.116.82.130
Connecting to mirrorblender.top-ix.org (mirrorblender.top-ix.org)|194.116.82.130
|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 249229883 (238M) [video/quicktime]
Saving to: `big_buck_bunny_480p_h264.mov'

66% [=======================>             ] 164,606,720 1.72M/s  eta 33s
```

图 12-3 下载测试用的影片

5. 执行

通过 HDMI 的屏幕和键盘,并输入命令来执行结果,如图 12-4 所示。注意在 SSH 上是看不到效果的。

```
$ mplayer big_buck_bunny_480p_h264.mov
```



图 12-4　执行结果

6. 教学影片

完整的教学视频可以看 RaspberryPi_Mplayer。

12.3　视频流服务器——VLC

案例 73

1. 介绍

在案例将会介绍同样鼎鼎大名的多媒体软件 VLC,它除了可以当做播放器使用,还可以做多媒体的格式转换,并具有广播的功能。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络连接设备。

会安装以下软件:

(1) Raspivid;

(2) VLC。

3. 接线

确认树莓派的网络连接功能正常。

4. 步骤

STEP1:升级安装程序

在安装 VLC 之前,更新系统的安装程序 apt-get。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

STEP2:安装 VLC

输入以下的命令,在树莓派上安装 VLC,如图 12-5 所示。

```
$ sudo apt-get install vlc
```

```
pi@raspberrypi ~ $ sudo apt-get install vlc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
```

图 12-5　安装 VLC

5. 结果

输入以下命令,下载测试用的影片,如图 12-6 所示。

```
$ wget http://mirrorblender.top-ix.org/peach/bigbuckbunny_movies/big_buck_bunny_480p_
h264.mov
```

```
pi@raspberrypi ~ $ wget http://mirrorblender.top-ix.org/peach/bigbuckbunny_movie
s/big_buck_bunny_480p_h264.mov
```

图 12-6　下载测试用的影片

输入以下命令播放文件 big_buck_bunny_480p_h264.mov,如图 12-7 所示。

```
$ vlc big_buck_bunny_480p_h264.mov
```



图 12-7　播放效果

注意　播放 my_file.mpg 文件。

```
$ vlc -vvv my_file.mpg
```

播放 http://www.example.org/your_file.mpg 文件。

```
vlc -vvv http://www.example.org/your_file.mpg
```

播放 RTSP://www.example.org/stream 的影像流媒体。

```
vlc -vvv rtsp://www.example.org/stream
```

播放 UDP 局域网中的 server_port 的广播影片。

```
vlc -vvv udp:[@:server_port]
```

将 a1.wmv 转换为 h.264 格式,并存储文件名为 a2.mp4。

```
vlc" -I dummy -vvv a1.wmv -- sout = #transcode{vcodec = "h264",vb = "512",fps = "23.97",
scale = "1",acodec = "mpga", ab = "128","channels = 2", samplerate = "44100"}:standard
{access = "file",mux = "dummy",dst = "a2.mp4"}
```

播放 big_buck_bunny_480p_h264.mov,把影片通过端口 8160 传到局域网,并使用 h264 格式。

```
$ vlc -vvv  big_buck_bunny_480p_h264.mov  -- sout '#standard{access = http,mux =
ts,dst = :8160}' :demux = h264
```

播放 big_buck_bunny_480p_h264.mov,把影片通过端口 8080 传到局域网,并使用 mms 格式给 Windows Media Player。

```
$  vlc -vvv input_stream -- sout '#transcode{vcodec = DIV3,vb = 256,scale = 1,
acodec = mp3,ab = 32,channels = 2}:std{access = mmsh,mux = asfh,dst = :8080}'
```

把网络上的音乐播放清单录下到 NPR-test.mp3

```
$ vlc "http://www.npr.org/streams/mp3/nprlive24.pls" -- sout file/mp3:NPR-test.mp3
```

12.4 获取和设置 Repository/Git /Github 版本控制

案例 74

1. 介绍

本案例介绍如何设置 Linux 支持 Git 的版本管理,方便让开发者做程序版本控制,并且网络上很多树莓派的原始代码都放在 github 上面。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络连接设备。

学习以下软件:

Git

3. 步骤

STEP1:设置 github.com 的账号

如果没有 github.com 的账号与密码,首先通过浏览器连接到 https://github.com 注册一组账号,如图 12-8 所示。



图 12-8 到 github.com 注册一组账号

STEP2：安装 Git 版本控制软件

在安装 Git 之前，更新系统的安装程序 apt-get。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

接下来安装 Git，如图 12-9 所示。

```
$ sudo apt - get install git
```

```
pi@raspberrypi ~/code/01httpweather $ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version.
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

图 12-9 安装 Git 版本控制软件

STEP3：初次设置安装 Git

在根目录创建一个 samplecode 的路径，通过 init 初始化的参数，设置 Repository 的工作路径。

```
$ cd ~
$ mkdir samplecode
$ cd samplecode
$ git init
```

接下设置 Repository 的用户名称和 EMAIL 相关数据。

```
$ git config -- global user.name "Your Name Here"
```

```
$ git config -- global user.email "your_email@example.com"
```

STEP4：下载 github 上面的源程序

到 github 挑选喜欢的源程序，通过以下命令下载 github 上面的源程序，例如本书在 github.com 上面的开放样例程序可以通过以下命令获取，如图 12-10 所示。

```
$ git clone https://github.com/powenko/RaspberryPi.git
```

取得完整的样例程序，然后通过 ls 查看到相关文件。

```
pi@raspberrypi ~/samplecode $ git clone https://github.com/powenko/RaspberryPi.g
it
Cloning into 'RaspberryPi'...
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 12 (delta 2), reused 12 (delta 2)
Unpacking objects: 100% (12/12), done.
pi@raspberrypi ~/samplecode $ ls
RaspberryPi
```

图 12-10　获取 github 原始程序

STEP5：程序版本管理 Repository 修改记录

为了体现出 Git 有能力做程序版本管理 Repository，在这里读者可以新增或者修改文件，例如修改 4.txt 的数据：

```
$ sudo nano 4.txt
```

通过按下键盘 Ctrl+O 组合键存储程序，Ctrl+X 组合键离开文本编辑器 nano。

接下来增加一个版本控制的记录点，这个记录点在这台树莓派的机器上，输入以下命令添加文件。

```
$ git add 4.txt
```

如果有多个文件，可以通过以下命令添加所有的文件。

```
$ git add .
```

编写注释和增加一个版本控制的记录点，如图 12-11 所示。

```
$ git commit - am " add 3.txt"
```

```
pi@raspberrypi ~/samplecode/RaspberryPi $  sudo nano 3.txt
pi@raspberrypi ~/samplecode/RaspberryPi $
pi@raspberrypi ~/samplecode/RaspberryPi $ git add .
pi@raspberrypi ~/samplecode/RaspberryPi $ git commit -am " add 3.txt"
[master 0ffe443]  add 3.txt
 1 file changed, 1 insertion(+)
 create mode 100644 3.txt
```

图 12-11　增加一个版本控制的记录点

STEP6：查看程序版本管理 Repository 历史记录

版本控制最重要的是可以了解团队的程序修改记录，可以输入以下命令查看，如图 12-12 所示。

```
pi@raspberrypi ~/samplecode/RaspberryPi $ git log
commit c67f5d81d1b6993d6364917034e95c8fef043cc5
Author: powen ko
Date:   Thu May 1 10:28:56 2014 -0700

    add powenko.txt

    add powenko.txt

commit 6beddaa681de36a9a0ace7faee30a3075d115f80
Author: powen ko
Date:   Thu May 1 10:28:07 2014 -0700

    1

    1

commit 0f56514d94375377bddec936e856346a36c53252
Author: powenko
```

图 12-12　通过 git log 获得到历史记录

```
$ git log
```

STEP7：退回 Repository 版本

如果想退回到之前的 Repository 版本，可以通过以下命令查看。

```
$ git checkout xxx
```

xxx 可以由 git log 中获取，例如

```
$ git checkout 0ffe44388fb182cad535ca6a7bcf447804b83df5
```

这样就可以随时退回去之前的版本，如图 12-13 所示。

```
pi@raspberrypi ~/samplecode/RaspberryPi $ git checkout 0ffe44388fb182cad535ca6a7
bcf447804b83df5
Note: checking out '0ffe44388fb182cad535ca6a7bcf447804b83df5'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

HEAD is now at 0ffe443...  add 3.txt
pi@raspberrypi ~/samplecode/RaspberryPi $ ls
01httpweather  1.txt  2.txt  3.txt
```

图 12-13　退回到之前的 Repository 版本

STEP8：与 github 数据同步

为了体现出 Git 有能力做团队合作与远程管理，试着把刚刚修改的数据送到 github.com。输入以下命令设置远程的网址。

```
$ git remote set-url origin https://github.com/powenko/RaspberryPi.git
```

再通过 push 的方法把数据送上 github。

```
$ git push origin master
```

输入在 github. com 上的账号密码就成功了,如图 12-14 所示。

```
pi@raspberrypi ~/samplecode/RaspberryPi $ git  remote set-url origin  https://gi
thub.com/powenko/RaspberryPi.git
pi@raspberrypi ~/samplecode/RaspberryPi $ git push origin master
Username for 'https://github.com': powenko
Password for '                          ':
Counting objects: 7, done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 539 bytes, done.
Total 6 (delta 1), reused 0 (delta 0)
To https://github.com/powenko/RaspberryPi.git
   c37a5ee..0f56514  master -> master
pi@raspberrypi ~/samplecode/RaspberryPi $ ls
01httpweather  1.txt  2.txt  3.txt  4.txt
```

图 12-14　送到 github. com

4. 结果

通过网页来确认,如图 12-15 所示。

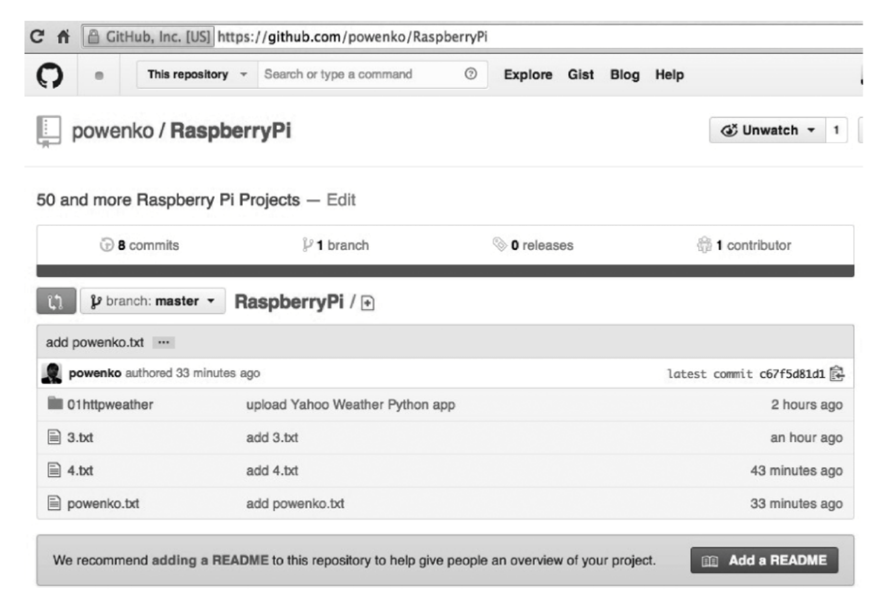


图 12-15　更新后的 github. com

5. 教学影片

完整的教学视频可以看 RaspberryPi_02GitHub. mov。

注意　当把程序和数据放到 github. com 上就会发生一个特别的状况,就是这个应用同时会被很多人编写和修改。如果再修改时,想获取别人已经更新后的数据,可以输入以下命令,获取网上的最新版本,如图 12-16 所示。

```
$ git pull
```

```
pi@raspberrypi ~/samplecode/RaspberryPi $ git pull
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 2), reused 3 (delta 1)
Unpacking objects: 100% (4/4), done.
From https://github.com/powenko/RaspberryPi
   0f56514..c67f5d8  master     -> origin/master
Updating 0f56514..c67f5d8
Fast-forward
 1.txt       |    3 ---
 2.txt       |    4 ----
 powenko.txt |    2 ++
 3 files changed, 2 insertions(+), 7 deletions(-)
 delete mode 100644 1.txt
 delete mode 100644 2.txt
 create mode 100644 powenko.txt
pi@raspberrypi ~/samplecode/RaspberryPi $ ls
01httpweather  3.txt  4.txt  powenko.txt
```

图 12-16 获取网络最新版本

获取网络最新版本的完整教学视频可以看 RaspberryPi_02GitHub_pull。

第 13 章 安全监视系统

13.1 影像监视与动作捕捉

案例 75

1. 介绍

本案例将会用树莓派板子上内置 Camera 做一个监视系统，但是与“11.2 树莓派摄像机”有何不同呢？这次会增加影响影像判断的功能，也就是说当画面不一样时才会引导摄像的动作，这样可以节省存储空间。

2. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 网络连接设备；

(3) 内接式的照相机。

3. 接线

把内接式的照相机接到树莓派开发板的 CSI 接口上。

4. 步骤

STEP1：安装所需的软件

输入以下命令，如图 13-1 所示，安装一些需要的软件。

```
$ sudo apt-get install cmake libopencv-dev
```

```
pi@raspberrypi ~ $ sudo apt-get install cmake libopencv-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
```

图 13-1 安装 libopencv-dev

STEP2：下载 userland 的源程序

输入以下命令，如图 13-2 所示，安装一些需要的软件。

```
$ mkdir －p /home/pi/src/raspberrypi
$ cd /home/pi/src/raspberrypi
$ git clone －－depth 1 https://github.com/raspberrypi/userland.git
```

```
pi@raspberrypi ~ $  mkdir -p /home/pi/src/raspberrypi
pi@raspberrypi ~ $ cd /home/pi/src/raspberrypi
pi@raspberrypi ~/src/raspberrypi $  git clone --depth 1 https://github.com/raspb
errypi/userland.git
Cloning into 'userland'...
remote: Counting objects: 669, done.
remote: Compressing objects: 100% (536/536), done.
remote: Total 669 (delta 178), reused 403 (delta 106)
Receiving objects: 100% (669/669), 31.24 MiB | 820 KiB/s, done.
Resolving deltas: 100% (178/178), done.
Checking out files: 100% (562/562), done.
```

图 13-2　下载 userland 的源程序

STEP3：编译程序

输入以下命令，如图 13-3 所示，安装一些需要的软件。

```
$ make －C /opt/vc/src/hello_pi/libs/vgfont
```

```
pi@raspberrypi ~/src/raspberrypi $  make -C /opt/vc/src/hello_pi/libs/vgfont
make: Entering directory `/opt/vc/src/hello_pi/libs/vgfont'
cc -D_HAVE_TIMER_T -DSTANDALONE -D__STDC_CONSTANT_MACROS -D__STDC_LIMIT_MACROS -
DTARGET_POSIX -D_LINUX -fPIC -DPIC -D_REENTRANT -D_LARGEFILE64_SOURCE -D_FILE_OF
```

图 13-3　编译程序

STEP4：编译

输入以下方法进行编译，如图 13-4 所示。

```
$ cd userland
$ mkdir build
$ cd build
$ cmake ../
```

```
pi@raspberrypi ~/src/raspberrypi $ ls
userland
pi@raspberrypi ~/src/raspberrypi $ cd userland
pi@raspberrypi ~/src/raspberrypi/userland $ mkdir build
pi@raspberrypi ~/src/raspberrypi/userland $ cd build
pi@raspberrypi ~/src/raspberrypi/userland/build $ cmake ../
-- The C compiler identification is GNU 4.6.3
-- The CXX compiler identification is GNU 4.6.3
-- Check for working C compiler: /usr/bin/gcc
-- Check for working C compiler: /usr/bin/gcc -- works
```

图 13-4　准备编译

输入 make 命令，编译程序，如图 13-5 所示。

```
$ make
```

```
pi@raspberrypi ~/src/raspberrypi/userland/build $ make
Scanning dependencies of target vcos
[  1%] Building C object interface/vcos/pthreads/CMakeFiles/vcos.dir/vcos_pthrea
ds.c.o
[  1%] Building C object interface/vcos/pthreads/CMakeFiles/vcos.dir/vcos_dlfcn.
c.o
```

图 13-5　编译

STEP5：下载 rpi-mmal-opencv-modetect 源程序

接下来下载 rpi-mmal-opencv-modetect 源程序，如图 13-6 所示。

```
$ cd /home/pi/src/raspberrypi
$ wget https://github.com/sodnpoo/rpi - mmal - opencv - modetect/archive/master.zip
```

```
pi@raspberrypi ~/src/raspberrypi/userland/build $  cd /home/pi/src/raspberrypi
pi@raspberrypi ~/src/raspberrypi $ ls
userland
pi@raspberrypi ~/src/raspberrypi $ wget https://github.com/sodnpoo/rpi-mmal-open
cv-modetect/archive/master.zip
```

图 13-6　下载 rpi-mmal-opencv-modetect 源程序

STEP6：解压缩

输入以下命令解压缩，如图 13-7 所示。

```
$ unzip master.zip
$ cd ~/src/raspberrypi/userland
$ mkdir build2
$ cd build2
$ cmake ../rpi - mmal - opencv - modetect - master
```

```
pi@raspberrypi ~/src/raspberrypi/userland/build $  cd /home/pi/src/raspberrypi
pi@raspberrypi ~/src/raspberrypi $ ls
userland
pi@raspberrypi ~/src/raspberrypi $ wget https://github.com/sodnpoo/rpi-mmal-open
cv-modetect/archive/master.zip
```

图 13-7　解压缩

STEP7：编译程序

进行程序编译，如图 13-8 所示。

```
$ make
```

```
pi@raspberrypi ~/src/raspberrypi/userland/build2 $ make
Scanning dependencies of target mmal_opencv_modect
[100%] Building C object CMakeFiles/mmal_opencv_modect.dir/opencv_modect.c.o
Linking C executable mmal_opencv_modect
[100%] Built target mmal_opencv_modect
```

图 13-8　编译程序

STEP8：打开 Camera 设置

为了打开 Camera 的设置，需要进入设置页。

```
$ sudo raspi - config
```

选择 Enable Camera（引导照相机），如图 13-9 所示。

通过 Tab 按键，选择 Enable 选项，按下 Enter 键回到设置主菜单后，再单击 Finish 按钮，重新引导树莓派。



图 13-9　选择 Enable Camera

5. 结果

输入以下命令来执行应用,并指定存储的文件名为 video. h264,如图 13-10 所示。

```
$ cd ~/src/raspberrypi/userland/build2
$ ./mmal_opencv_modect > video.h264
```

```
pi@raspberrypi ~/src/raspberrypi/userland/build2 $ ./mmal_opencv_modect > video.h264
VIDEO_WIDTH : 1280
VIDEO_HEIGHT: 720
VIDEO_FPS   : 30
```

图 13-10　执行程序

当摄像机发现画面不一样时,便会引导摄像的动作,如图 13-11 所示。

```
MOTION DETECTED (342)
MOTION DETECTED (1928)
MOTION DETECTED (1433)
MOTION DETECTED (199)
MOTION DETECTED (198)
MOTION DETECTED (200)
MOTION DETECTED (117)
FPS: OpenCV = 14.90, Video = 30.52
```

图 13-11　检测到画面不同

实际硬件的执行情况,如图 13-12 所示。

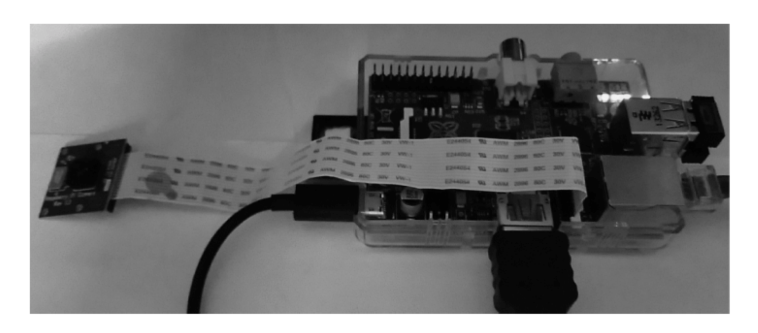


图 13-12　实际硬件的执行情况

6. 教学影片

完整的教学视频可以看 RaspberryPi_Camera_motionDetected。

13.2 树莓派实时影像流服务器

案例 76

1. 介绍

本案例将会用树莓派板子上的内置 Camera 接口来获取影像，通过网络 Streaming Protocol 的方法将摄像机的影像传到网络上。

2. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 网络连接设备；

(3) 内接式的照相机。

需要有以下软件：

(1) Raspivid；

(2) VLC。

3. 准备

本案例执行前需要先完成：

(1) 内置 Camera Board 拍照，请查看"11.1 树莓派照相机"。

(2) 内置 Camera Board 摄像，请查看"11.2 树莓派摄像机"。

4. 接线

把内接式的照相机接到树莓派开发板的 CSI 接口上。

5. 步骤

STEP1：升级安装程序

在安装 VLC 之前，更新系统的安装程序 apt-get。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

STEP2：确认打开摄像机

输入以下命令进入设置，并确认摄像机打开。

```
$ sudo raspi - config
```

选择 Enable Camera 来引导树莓派的摄像机，如图 13-13 所示。

STEP3：安装 VLC

输入以下命令在树莓派上安装 VLC，如图 13-14 所示。

```
$ sudo apt - get install vlc
```



图 13-13 引导摄像机

```
pi@raspberrypi ~ $ sudo apt-get install vlc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
```

图 13-14 安装 VLC

STEP4：执行 streaming

输入以下命令，就可以开始制作视频流，如图 13-15 所示。

```
$ raspivid -o -- t 0 - hf - w 800 - h 400 - fps 24 |cvlc - vvv stream:///dev/stdin -- sout '
#standard{access = http,mux = ts,dst = :8160}' :demux = h264
```

参数介绍和设置：

（1）w：视频宽度；

（2）h：视频高度；

（3）fps：每秒传输帧数；

（4）dst：网络的端口；

（5）demux：h264 视频压缩格式。

```
pi@raspberrypi: ~
pi@raspberrypi ~ $ raspivid -o - -t 0 -hf -w 800 -h 400 -fps 24 |cvlc -vvv strea
m:///dev/stdin --sout '#standard{access=http,mux=ts,dst=:8160}' :demux=h264
```

图 13-15 开始制作视频流

6. 结果

执行后会看到摄像机拍摄的视频显示在画面上,并且可以在其他平台通过 VLC 软件连接到树莓派上观看摄像机的实时影像,效果如图 13-16 所示。

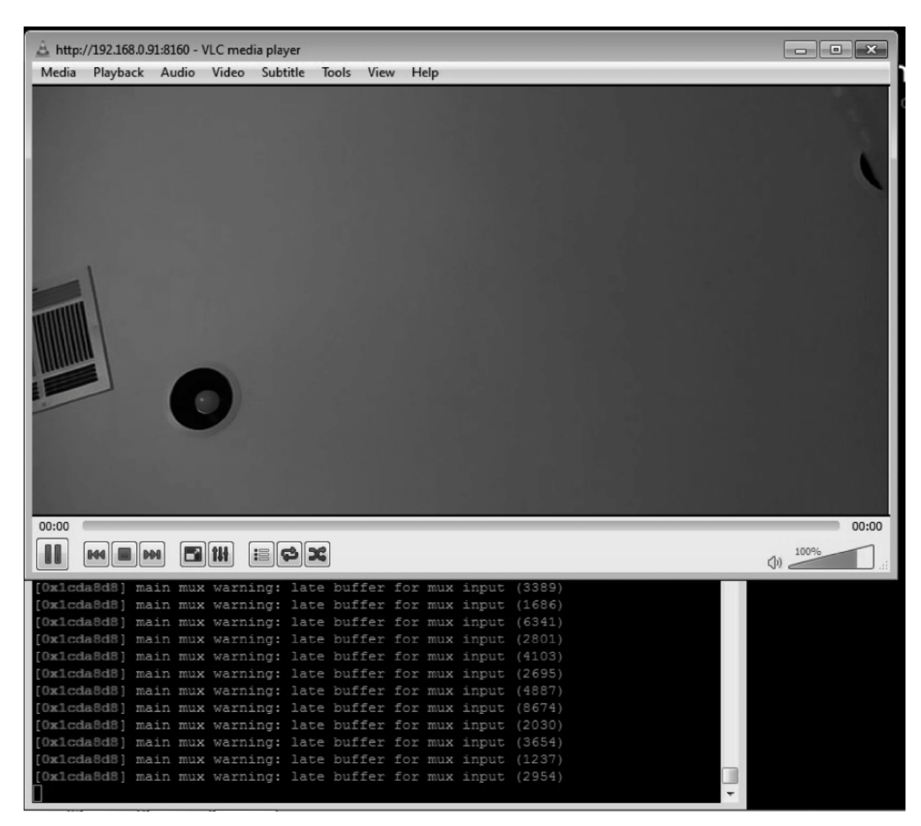


图 13-16 执行效果

连接地址 http://树莓派的 IP 地址:8160/。

7. 教学影片

完整的教学视频可以看 RaspberryPi_VLC_streaming。

注意 1) 开机自动引导

如果想要开机就能自动引导,请按照以下步骤设置树莓派。

通过文本编辑器 nano 创建/etc/init.d/webcamstart.sh 文件。

```
$ sudo nano /etc/init.d/webcamstart.sh
```

并把以下内容写入文件,如图 13-17 所示,完成后按下 Ctrl+O 组合键存储,Ctrl+X 组合键离开文本编辑器。

```
GNU nano 2.2.6          File: /etc/init.d/webcamstart.sh             Modified

#!/bin/sh
$///dev/stdin --sout '#standard{access=http,mux=ts,dst=:8160}' :demux=h264

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 13-17　/etc/init. d/webcamstart. sh 内容

```
#!/bin/sh
sleep 60; raspivid -o -- t 0 -hf -w 800 -h 400 -fps 24 |cvlc -vvv stream:///dev/
stdin -- sout '#standard{access = http,mux = ts,dst = :8160}' :demux = h264
```

修改 Shell Script 文件权限。

```
$ sudo chmod 755 /etc/init.d/webcamstart.sh
```

更新开机配置文件就完成了。

```
$ sudo update-rc.d webcamstart.sh defaults
```

2) 补充数据

如果要用 RTSP 的方法传递数据,可以输入以下命令完成。

```
$ raspivid -o -- t 0 |cvlc -v stream:///dev/stdin -- sout '#rtp{sdp = rtsp://
:8160/}' :demux = h264
```

完成后,可以在其他平台通过 VLC 软件连接到树莓派上观看摄像机的实时影像,效果如图 13-18 所示。

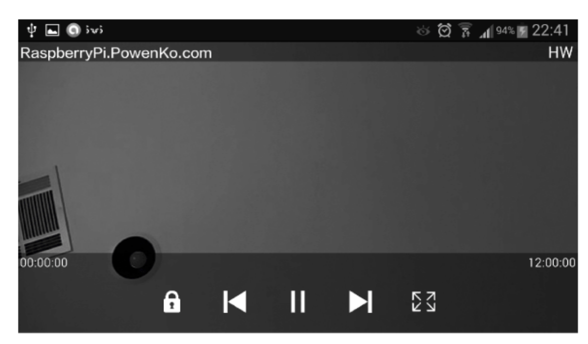


图 13-18　在 Android 手机上看到树莓派的实时视频

连接地址 rtsp://树莓派的 IP 位置:8160/。

如果要用 M3U8 的方法传递数据,可以通过以下命令完成。M3U8 是什么格式呢?是 iOS 平台 MediaPlayer Framework 的实时多媒体影音流的格式,也是 iOS 平台当前唯一可以观看实时影音的格式。

```
$ raspivid -o --t 0 | cvlc -v -I "dummy" stream:///dev/stdin :sout = "#std{access = livehttp{seglen = 10,delsegs = true,numsegs = 5,index = /var/www/streaming/stream.m3u8,index-url = http://树莓派的 IP 地址/streaming/stream-########.ts},mux = ts{use-key-frames},dst = /var/www/streaming/stream-########.ts}" :demux = h264
```

iOS 开发者在 MediaPlayer Framework 上指定以下 URL 就可以了。

```
http://树莓派的 IP 地址/streaming/stream.m3u8
```

13.3 数字监视播放器——在计算机上监视

案例 77

1. 介绍

VLC 在每个平台上都有适配的软件,例如 Windows、Linux 和 Mac 平台,只要通过网页连接到 http://www.videolan.org/vlc/,如图 13-19 所示,就可以下载该软件,因为各平台安装与使用方法几乎一样,这里就以 Windows 为例来介绍如何设置与连接到树莓派影像流服务器。

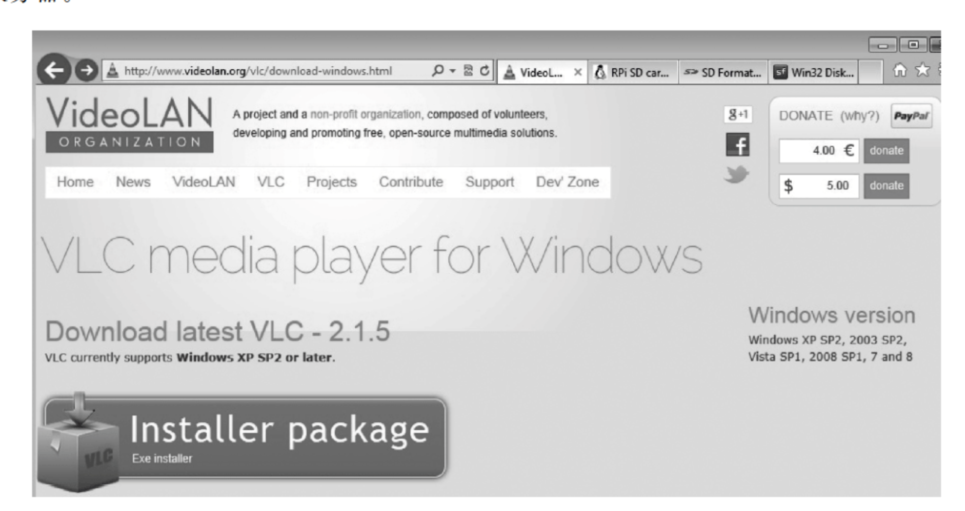


图 13-19 下载 Windows 版本的 VLC 软件

2. 播放树莓派实时视频

下载后安装、执行,并选择 Open Network Stream,如图 13-20 所示。



图 13-20　在 VLC 选择 Open Network Stream

输入树莓派实时视频的网址,例如 http://树莓派的 IP 地址:8160/。

单击 Play 播放,如图 13-21 所示。



图 13-21　在输入 URL 网址后单击 Play 播放

可以看到树莓派上的视频,如图 13-22 所示,因为视频缓冲的原因,会有 2 秒钟的延迟。



图 13-22 看到树莓派上的视频

13.4 数字监视摄像机——在计算机上录制

案例 78

1. 目的

很多时候,需要在计算机上一边看、一边录制远程树莓派传递过来的视频,这样就能够将远程监控机作为摄像机。那么要如何设置呢?请按照以下的步骤处理。

2. 步骤

选择 VLC 的 Convert/Save,如图 13-23 所示。

在 convert 窗口,如图 13-24 所示,设置以下属性:

(1) Source:输入树莓派实时视频的网址,例如 http://树莓派的 IP 位置:8160/。

(2) Display the output:是否要显示播放内容,作勾选。

(3) Destination file:存储的文件位置,输入地址。

单击 Start 按钮开始录制。



图 13-23　选择 VLC 的 Convert/Save…

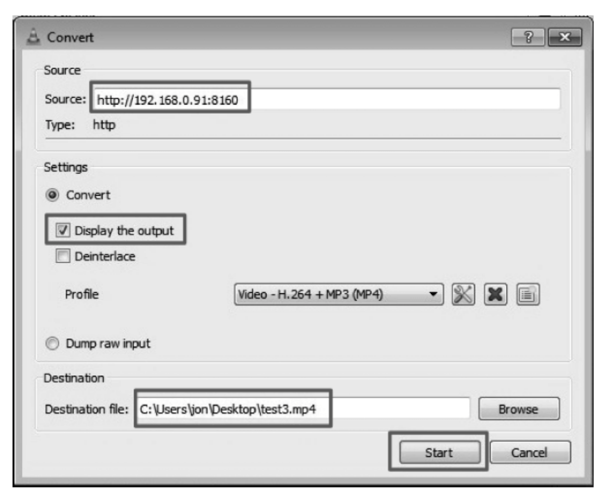


图 13-24　设置

13.5　在 iOS 和 Android 上观看实时视频

案例 79

1. Android 操作系统

VLC 在 Android 平台上有适配的软件，在 Google Play 上查找 VLC 就可以免费下载，如图 13-25 所示，输入树莓派实时视频的网址，单击 Play 按钮就能够顺利播放。

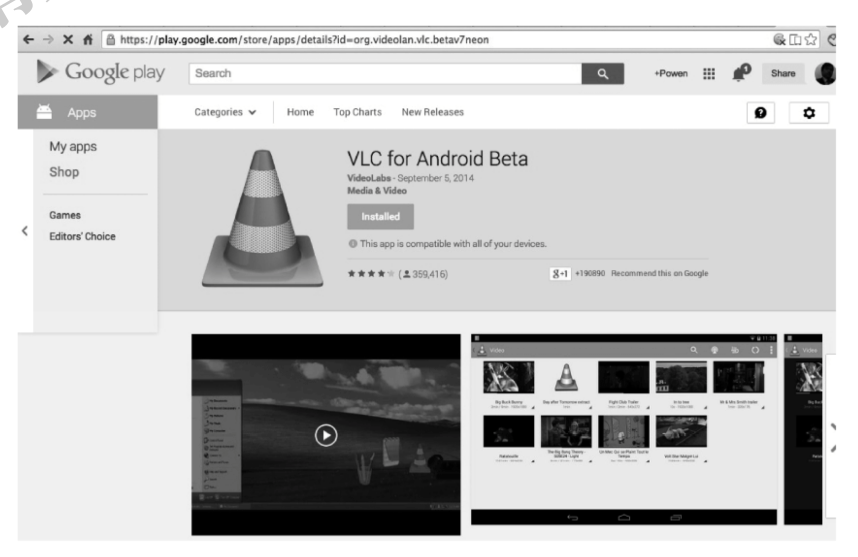


图 13-25　Android 平台的 VLC 软件

2. iOS 操作系统

VLC 在 iOS 平台上也有适配的软件，在 App Store 上查找 VLC 就可以免费下载，如图 13-26 所示，输入树莓派实时视频的网址，单击 Play 按钮就能够顺利播放。

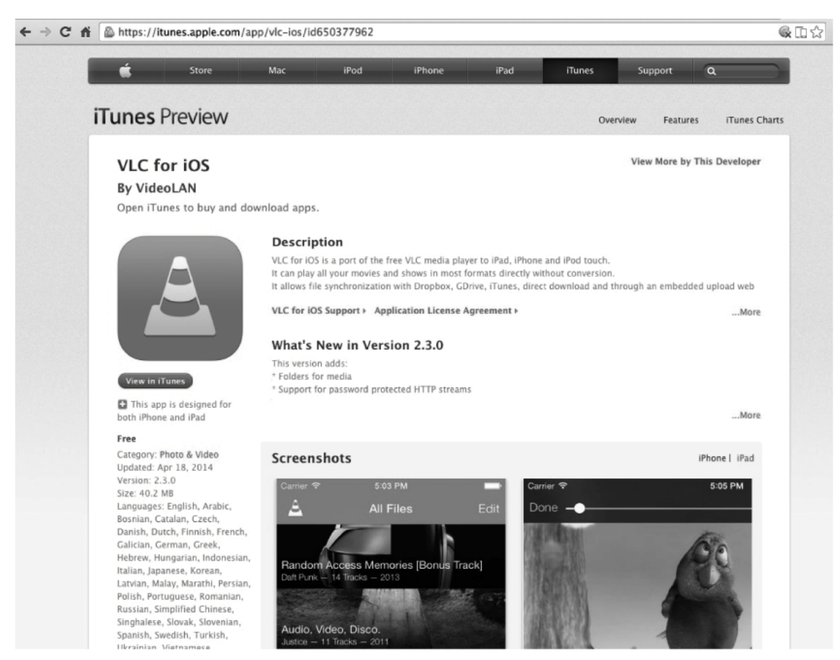


图 13-26　iOS 平台的 VLC 软件

第 14 章

增强现实影像辨识系统

通过摄像机实时捕捉影像，并且做影像辨识判断，在屏幕上把虚拟世界与现实世界进行叠加互动，打造增强现实(Augmented Reality，AR)的效果。

14.1 显示和捕获 USB Webcam 的画面

案例 80

1. 介绍

本案例通过 USB 的网络摄像机拍摄画面，通过树莓派显示影像，并且进一步把显示出来的画面存到文件中。整个过程会用 Python 程序语言来完成，用来拼接下一个增强现实的应用。

2. 硬件规范

本案例会用到 USB 的网络摄像机，但是当前树莓派可以支持的 USB Webcam 网络摄像机没有几个，请挑选合适的品牌，如图 14-1 所示，不然 Linux 会找不到驱动程序，当前确定可以正常工作的有：

(1) Logitech HD Webcam C270；

(2) Creative Go PD00040；

(3) Creative Live! Cam Vista IM VF0640；

(4) 微软 NX-6000；

(5) Creative Carl Zeiss。

详细请参考官方的视频摄像机兼容列表 http://elinux.org/RPi_USB_Webcams。

3. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 1 个 USB 的网络摄像机。

需要有以下软件：

(1) Python；



图 14-1　微软 NX-6000 和 Creative Carl Zeiss

(2) Python GPIO 函数库。

4. 接线

将 USB 网络摄像机接到树莓派的 USB 接口,如图 14-2 所示。

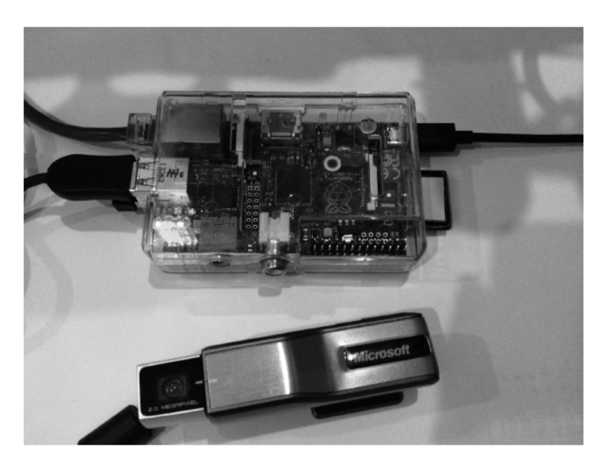


图 14-2　将 USB 摄像机接到树莓派的 USB 接口

5. 步骤

STEP1:更新安装软件

需要从网络上下载最新版本的软件,所以请先更新 apt-get。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

STEP2:安装 USB 摄像机软件

需要安装一套叫做 LUCView 的软件,输入以下命令就可以获取该软件。

```
$ sudo apt-get install luvcview
```

STEP3：测试 USB 摄像机软件 LUCView

安装成功后，执行以下命令测试 USB 摄像机是否可以正常工作，这里参数-s 是设置要显示的屏幕尺寸，如果不指定，默认尺寸是 640×480。

```
$ luvcview -s 320x240
```

注意　执行这个程序命令时，请通过 HDMI 接到电视或者屏幕上才会看到输出影像。

---

6. 结果

可以进入窗口模式或者一般的文本模式执行 LUCView，看到 USB 网络摄像机的画面显示在屏幕上，如图 14-3 所示。

```
$ luvcview
```

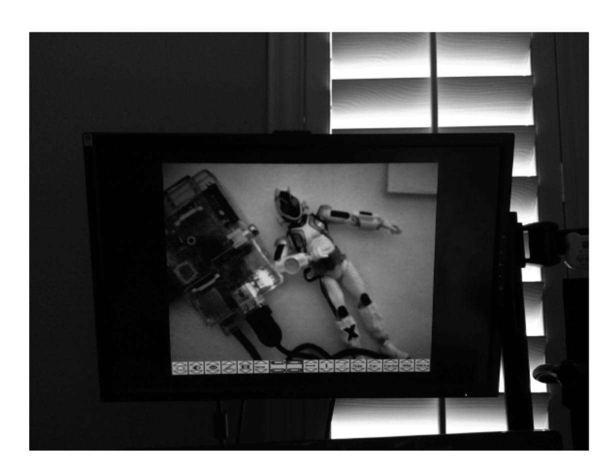


图 14-3　LUCView 执行后的情况

7. 教学影片

完整的教学视频可以看 RaspberryPi_USBWebCam_1_Display。

14.2　Python 图形处理函数库——SimpleCV，OpenCV

案例 81

1. 介绍

本案例目的是使用 Python 语言编写程序显示一张图片，并且在这张图片上添加一些文字或者另外图片的效果。

为了达到目的,需要使用 SimpleCV 函数库,SimpleCV 的功能非常多,如表 14-1 所示,SimpleCV 其实就是通过 OpenCV 让开发者更容易地在 Python 开发图形处理的函数库。进一步学习可登录官网 http://tutorial.simplecv.org。修改本案例可以看到更多有趣的效果。

表 14-1 常用的图形函数

样　例	功　能
Image1.scale(90,90)	图片尺寸调整
Image1.crop(100,100,50,50)	画面剪切
Image1.save("powenko.png")	图片存档
Image1.show()	图片显示

2. 硬件规范

同案例 80。

3. 步骤

先执行案例 80,再继续以下的操作。

STEP1:安装 python-opencv

需要通过 Python 程序语言来做图形处理,这时 SimpleCV 函数库就派上用场了,这个开源代码重要的功能是做影像显示、图片处理和图形文件格式转换等。可以通过以下命令取得相关函数库,方法如图 14-4 所示,整个下载时间大约 5 分钟,请耐心等待。

```
$ sudo apt - get install python - opencv python - scipy python - numpy python - pip
```

```
pi@raspberrypi ~ $ sudo apt-get install python-opencv python-scipy python-numpy pyth
on-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-numpy is already the newest version.
python-numpy set to manually installed.
The following extra packages will be installed:
  libamd2.2.0 libavcodec53 libavformat53 libavutil51 libdc1394-22
```

图 14-4　python-opencv 实际安装情况

STEP2:下载和安装 SimpleCV

需要到 Github 下载源程序,如图 14-5 所示,可以通过 pip 来获取源程序,并下载和安装。

```
$ sudo pip install https://github.com/ingenuitas/SimpleCV/zipball/master
```

```
pi@raspberrypi ~/code/cv $ sudo pip install https://github.com/ingenuitas/SimpleCV/z
ipball/master
Downloading/unpacking https://github.com/ingenuitas/SimpleCV/zipball/master
  Downloading master (43.8Mb): 43.8Mb downloaded
  Running setup.py egg_info for package from https://github.com/ingenuitas/SimpleCV/
zipball/master

Installing collected packages: SimpleCV
  Running setup.py install for SimpleCV

    Installing simplecv script to /usr/local/bin
Successfully installed SimpleCV
Cleaning up...
```

图 14-5　到 Github 下载和安装 SimpleCV

STEP3：测试 SimpleCV 是否安装成功

可以通过进入 Python 程序语言执行 SimpleCV，命令如图 14-6 所示，如果有错误，Python 就会显示找不到该函数库。

```
$ python
  >>> import SimpleCV
  >>> exit()
```

```
pi@raspberrypi ~/code/cv $ python
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import SimpleCV
>>> exit()
pi@raspberrypi ~/code/cv $
```

图 14-6 进入 Python 程序语言执行 SimpleCV

STEP4：编写程序

学习如何在 Python 的程序语言中显示图片，打开文本编辑器输入以下程序。

```
$ sudo nano displayimahe.py
```

样例程序：Sample\ch09\displayimahe . py

```
1.  from SimpleCV import Image,Display
2.  from time import sleep
3.  Display1 = Display()
4.  Image1 = Image("raspberrypi.png")
5.  Image1.save(Display1)
6.  while not Display1.isDone():
7.      sleep(1)
```

程序解说

第 1 行：指定 SimpleCV 函数库，并且设置类型为 Image 和 Display。

第 4 行：读入文件图片，并且指定到类型变量 Image1 上。

第 5 行：存储图片。

第 6 行：显示屏幕画面上。

4. 结果

在执行这个程序时，需要有张图片 raspberrypi. png 放在相同的路径中，可以通过以下命令，从网络上获取图片。

```
$ wget http://www.powenko.com/download_release/raspberrypi.png
```

执行 Python 程序时，可以输入以下命令。

```
$ python displayimahe .py
```

如果一切顺利,在纯文本模式上执行,就会在屏幕上看到的如图 14-7 的执行效果。

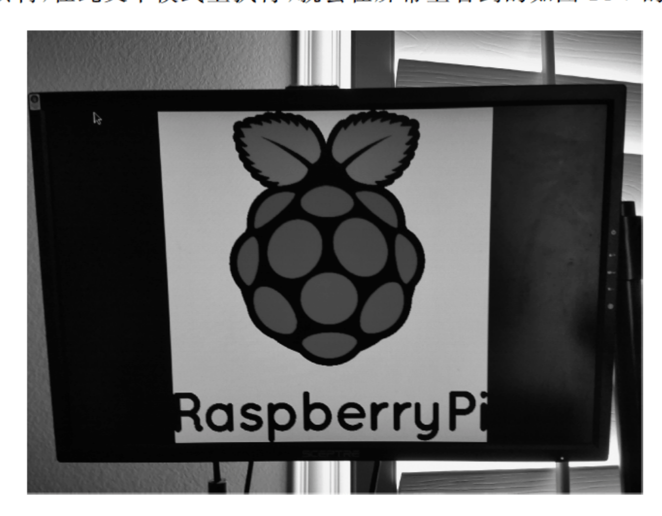


图 14-7 displayimahe.py 在纯文本模式上的执行效果

同样的操作,如果在窗口模式上执行,执行效果就会如图 14-8 所示。

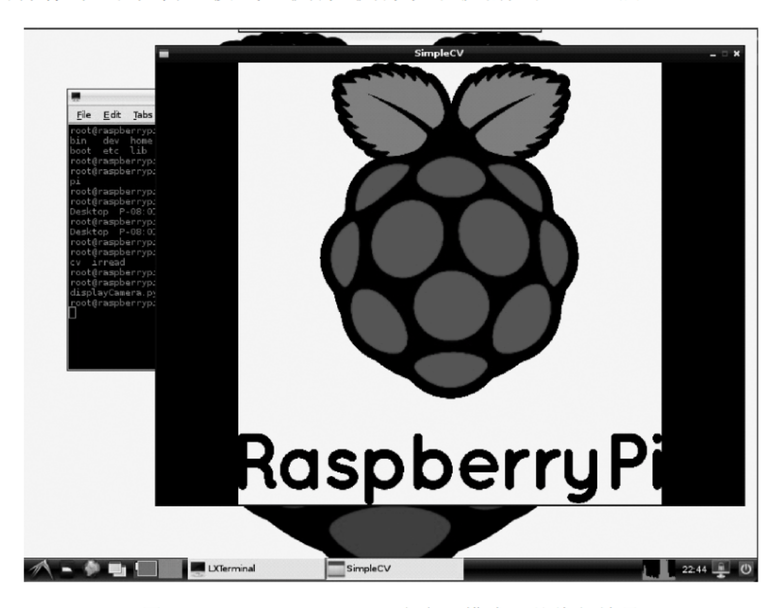


图 14-8 displayimahe.py 在窗口模式上的执行效果

5. 教学影片

完整的教学视频可以看 RaspberryPi_USBWebCam_2_install_SimpleCV。

14.3　人脸识别

案例 82

1. 介绍

本案例继续使用相同的函数库 SimpleCV 来达到影像识别功能。其实为何 SimpleCV 函数库可以识别人脸,仔细看一下,在下载文件 SimpleCV/Features/HaarCascades 里面有一大堆判断数据。如表 14-2 所示是 SimpleCV 中 XML 的功用。

表 14-2　可以辨识的范围和使用方法

文件名称	功能	使 用 方 法
eye. xml	眼睛的判断数据	findHaarFeatures('eye')
face. xml	脸部的数据	findHaarFeatures('face')
fullbody. xml	人全身的数据	findHaarFeatures('fullbody')
glasses. xml	眼镜的判断数据	findHaarFeatures('glasses')
left_ear. xml	左边耳朵的判断数据	findHaarFeatures('left_ear')
lefteye. xml	左边眼睛的判断数据	findHaarFeatures('lefteye')
lower_body. xml	下半身的判断数据	findHaarFeatures('lower_body')
mouth. xml	嘴巴的判断数据	findHaarFeatures('mouth')
nose. xml	鼻子的判断数据	findHaarFeatures('nose')
right_ear. xml	右边耳朵的判断数据	findHaarFeatures('right_ear')
righteye. xml	右边眼睛的判断数据	findHaarFeatures('righteye')
two_eyes_big. xml	两眼的判断数据	findHaarFeatures('two_eyes_big')
upper_body. xml	上半身的判断数据	findHaarFeatures('upper_body')

2. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 1 个 USB 的网络摄像机。

需要有以下软件：

(1) Python；

(2) Python GPIO 函数库。

3. 步骤

先执行案例 81,再继续以下的操作。

STEP1：编写程序

使用 Python 程序语言获取网络摄像机的影像,并且对影像的内容实时判断是否有人脸出现,所以需要使用 SimpleCV 函数库来编写程序,请打开文本编辑器输入以下程序,如图 14-9 所示。

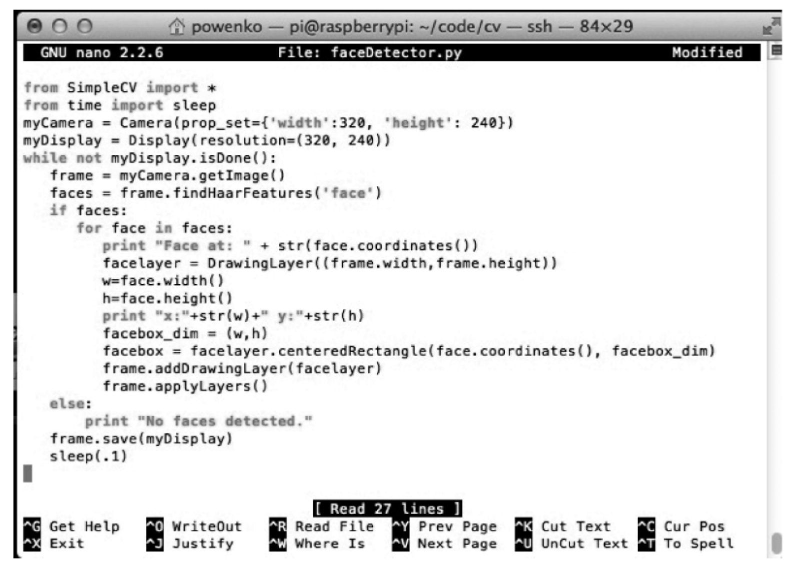


图 14-9　通过文本编辑器编写程序

```
$ sudo nano faceDectecor.py
```

样例程序：faceDectecor.py

```
1.  from SimpleCV import *
2.  from time import sleep
3.  myCamera = Camera(prop_set = {'width':320,'height': 240})
    # 指定网络摄像机的影像尺寸
4.  myDisplay = Display(resolution = (320,240))              # 显示窗口尺寸
5.  while not myDisplay.isDone():
6.    frame = myCamera.getImage()                             # 获取网络摄像机的影像
7.    faces = frame.findHaarFeatures('face')                  # 寻找人脸
8.    if faces:
9.      for face in faces:
10.         print "Face at: " + str(face.coordinates())       # 打印出人脸的中心位置
11.         facelayer = DrawingLayer((frame.width,frame.height))    # 准备一个新的图层
12.         w = face.width()
13.         h = face.height()
14.         print "x:" + str(w) + " y:" + str(h)
15.         facebox_dim = (w,h)
16.         facebox = facelayer.centeredRectangle(face.coordinates(),facebox_dim)
                                                              # 画出一个正方形
17.         frame.addDrawingLayer(facelayer)                  # 把图层放回画面上
18.         frame.applyLayers()
19.   else:
20.       print "No faces detected."
21.   frame.save(myDisplay)                                   # 显示在画面上
22.   sleep(.1)
```

程序解说

第 5 行：通过无穷循环一直处理摄像机的影像。

第 9～18 行：如果有人脸出现，便在图像上画一个框框圈着人脸。

第 21 行：显示屏幕画面上。

注意　程序的第 7 行 frame.findHaarFeatures('face') 用来识别脸部，可以参考表 14-2，修改成其他的识别部位，例如鼻子，就可以使用 findHaarFeatures('nose')。

---

4. 结果

执行 Python 程序，可以输入以下命令。

```
$ python faceDectecor.py
```

如果一切顺利，就会在屏幕上看到如图 14-10 所示的执行效果，系统将识别到的人脸用框框圈住。

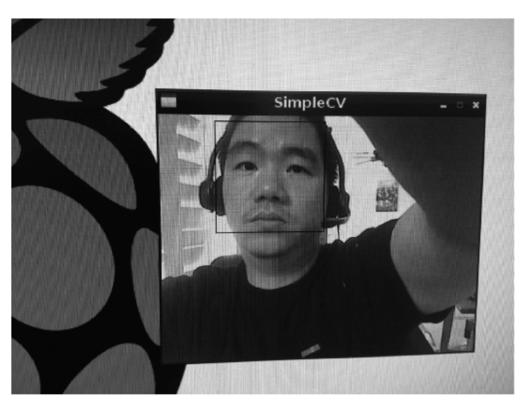


图 14-10　执行效果

5. 教学影片

完整的教学视频可以看 RaspberryPi_USBWebCam_3_Face_T。

注意　虽然识别花费的时间较长，但识别效果很惊人。建议修改程序调整成其他的识别效果，例如人物的辨识、眼睛的辨识等。

14.4　AR 增强现实

案例 83

1. 介绍

增强现实(Augmented Reality，AR)是一种实时地计算摄像机影像的位置及角度并加

上相应图像的技术,这种技术的目的是在屏幕上把虚拟世界与现实世界进行交互。这种技术在 1990 年提出,随着随身电子产品运算能力的提升,用途将会越来越广。

应当归为地图、GPS 地图导航的升级功能,应用 AR 技术将产品模型叠加到现实场景中。

以视觉为基础的增强现实是可以直接以摄像机做到识别和动作捕获,可以从街景深入到广告招牌、公车站牌、商店橱窗、店内陈设、商品、服务员的名牌、菜单等,并根据摄像机上的影像识别,实时叠加各类 3D 影片或信息内容。

有时人们会好奇增强现实或者实时动态影像处理到底是怎么做到的,在这个案例中会判断人的眼睛,并且在人的眼睛上面放上眼镜,看起来就像增强现实的效果。

2. 硬件规范

同案例 82。

3. 步骤

先执行案例 82,再继续以下的操作。

STEP1:准备要贴上的图片

准备一张单一近框的图片 glasses. png,必须有去背景的效果,也就是当自定义特别的颜色时,程序就不会把这个颜色画在屏幕上,可以造成一个中空去背景的效果,如图 14-11 所示。



图 14-11 准备要贴上的图片 glasses. png

这里指定的颜色 R,G,B 是(255,255,255)。

STEP2:编写程序

使用 Python 程序语言获取网络摄像机的影像,并且对影像的内容实时判断是否有人脸出现,所以需要使用 SimpleCV 函数库来编写程序,请打开文本编辑器输入以下程序,如图 14-12 所示。

```
$ sudo nano AR3.py
```

样例程序:AR3. py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
3.  from SimpleCV import *
4.  from time import sleep
5.  myCamera = Camera(prop_set={'width':320,'height': 240})  # 指定网络摄像机的影像尺寸
6.  myDisplay = Display(resolution=(320,240))               # 获取网络摄像机的影像
7.  pic = Image("glasses.png")
8.  mask2 = pic.createBinaryMask(color1=(0,0,0),color2=(254,254,254))
                                                          # 重点: 设置去背景颜色
9.  while not myDisplay.isDone():
10. frame = myCamera.getImage()
11. faces = frame.findHaarFeatures('face')                   # 寻找人脸
```

```
12.        if faces:
13.          for face in faces:
14.            print "Face at: " + str(face.coordinates())          # 打印出人脸的中心位置
15.            facelayer = DrawingLayer(frame.width,frame.height)   # 准备一个新的图层
16.            w = face.width()
17.            h = face.height()
18.            print "x:" + str(w) + " y:" + str(h)
19.            facebox_dim = (w,h)
20.            facebox = facelayer.centeredRectangle(face.coordinates(),facebox_dim)
                                                                    # 画出一个正方形
21.            frame.addDrawingLayer(facelayer)
22.            frame.applyLayers()                                  # 把图层放回画面上
23.            # 识别眼睛
24.            myFace = face.crop()                                 # 取脸部的影像
25.            eyes = myFace.findHaarFeatures('eye')                #识别眼睛
26.            if eyes:
27.                for eye in eyes:
28.                                                                 # 计算出眼镜的位置
29.                    xf = face.x - (face.width()/2)
30.                    yf = face.y - (face.height()/2)
31.                    xm = eye.x - (eye.width()/2)
32.                    ym = eye.y - (eye.height()/2)
33.                    x1 = pic.width
34.                    x1 = x1/2
35.                    x2 = eye.width()
36.                    x2 = x2/2
37.                    xmust = xf + xm - x1 + x2
38.                    ymust = yf + ym
39.                    print "Eye at: " + str(eye.coordinates())
40.                    frame = frame.blit(pic,pos = (xmust,ymust),mask = mask2)  #画出眼镜
41.        else:
42.          print "No faces detected."
43. frame.save(myDisplay)                    # 显示在画面上
44. sleep(.1)
```

程序解说

第 5 行：通过无穷循环一直处理摄像机的影像。

第 9～18 行：如果有人脸出现，便在图像上画一个框框圈着人脸。

第 25 行：判断眼睛。

第 28～39 行：计算出眼镜的位置。

第 40 行：画出眼镜。

第 21 行：显示屏幕画面上。

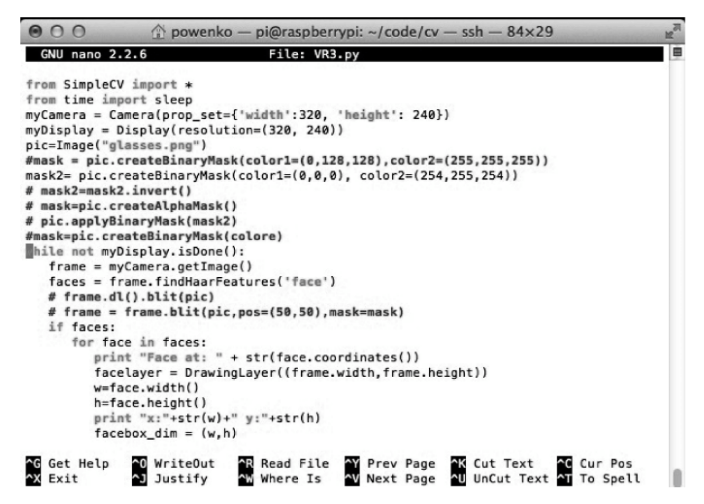


图 14-12 通过文本编辑器编写程序

注意 程序的第 8 行 pic.createBinaryMask(color1=(0,0,0),color2=(254,254,254))的功能是在这个范围之内的颜色会显示在画面上,所以全黑的颜色 R,G,B (255,255,255)就不会显示在屏幕上。

---

4. 结果

执行 Python 程序时,可以输入以下命令。

```
$ python AR3.py
```

如果一切顺利,就会在屏幕上看到如图 14-13 所示的执行效果,当识别到眼睛时便会画上眼镜的图片,然后识别到人脸就会在上面画一个框框圈住。



图 14-13 执行效果

5. 教学影片

完整的教学视频可以看 RaspberryPi_USBWebCam_4_AR_T。

第 15 章 条码扫描器

15.1 显示条码

案例 84

1. 介绍

条码(barcode)是一种代替键盘向计算机输入数字与字母的方法,只需扫描条码就可以将数字与字母输入到计算机中,无需通过键盘输入,即可以避免人为输入错误,也加快了数据输入时间。

条形码是将宽度不等的多个黑条和空白,按照一定的编码规则排列,用来表达一组信息的图形标识符,由读入这一条码的机器(扫描器)进行解读(解码),并传送和输入到计算机中。

条码按不同国家和组织的定义而有所不同,常见的种类如下:

(1) EAN 码(European Article Number)

由国际物品编码协会(International Article Numbering Association,IANA)制定的一种商品用条码,IANA 成立于 1977 年,初期以欧洲国家为主,EAN 码共有 13 位数字,由 0~9 组成,这些数字中有国家或地区码、厂商号码、产品编号及检查码。因此商品条形码可以说是任何国家、任何厂商以及任何商品独一无二的号码,也是国际市场中一种通行无阻的产品号码。

(2) UPC 码(Universal Product Code)国际产品码

1973 年,由美国制订,与 EAN 码同为当前全世界应用最广且统一规范的条码系统。UPC-A 码仅有 12 位数字,且彼此的数据内容排列顺序不同。每个字由 4 个直线条组成,其中两个暗线条,两个明线条。UPC 码的缩短码为 7 位数字,EAN 码则为 8 位数字。

(3) 39 码(Code 39)

1975 年,由美国 INTERMAKE 公司发表,比起 EAN 和 UPC 码,39 码广泛应用于制造业。它可代表 26 个英文字母、数字及特殊符号,并且可增加至美国信息交换标准码(ASC)所定的 128 个字。39 码是由 9 个直条组成,其中 5 个暗线条,4 个明线条;3 个宽条、6 个窄

条。39码是双向可读,且是包含的文字数字最多的条码系统。

其他条码如图15-1所示。



图15-1 由左至右分别是EAN-8、EAN-13、UPC-A条码

(1) JAN码:由EAN体系衍生出来的JAN码(Japanese Article Number)是日本于1978年发表的条码系统。

(2) 交叉25码:应用于汽车业、仓库业等。交叉25码是可采用文字密度较高的条码。

(3) CODE-BAR:应用于图书馆、医疗机构照片等。

(4) CODE-128:应用于企业内部管理,生产流程和物流控制系统等。

(5) ISBN码:应用于在图书的条码,前3位为978,书码9位,检核码1位。

(6) ISSN码:应用于杂志期刊,其前3位为977。

2. 目的

本案例介绍如何在屏幕上显示条码,也可以更进一步改良,将条码显示在液晶屏幕上。

3. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络设备。

需要有以下软件:

(1) Python;

(2) simpleCV函数库。

4. 步骤

STEP1:安装python-opencv

需要通过Python程序语言做图形处理和显示在画面上,这时SimpleCV函数库就派上用场了,按照以下命令安装SimpleCV。

```
$ sudo apt-get install python-opencv python-scipy python-numpy python-pip
```

需要到Github下载源程序,可以通过pip来获取源程序,并下载和安装。

```
$ sudo pip install https://github.com/ingenuitas/SimpleCV/zipball/master
```

STEP2:条码产生Web Service

通过http://www.powenko.com/webservice/barcode来显示条码,并获取和显示图

片,如图 15-2 所示,官方网站当前支持 UPC、EAN-8、EAN-13、UPC-E、S205、I2O5、I25、POSTNET、CODABAR、CODE128、CODE39 及 CODE93。

www.powenko.com/webservice/barcode/index.php

Select Encoding : UPC-A
Barcode Data : 120000000789
Barcode Height : 50
Scale : 2
Background Color : #FFFFEC
Bar Color : #333366
File Name* : PNG
Submit
* Give file name if you want to save the barcode else leave blank.

图 15-2 显示条码

本案例会通过 Web Service 的方法显示图片,并且获取下载,如图 15-3 所示。

```
http://www.powenko.com/webservice/barcode/index.php?onlyimage = yes&encode = UPC - A&bdata = 123456789012&height = 50&scale = 2&bgcolor = % 23FFFFEC&color = % 23000000&file = &type = png&Genrate = Submit
```

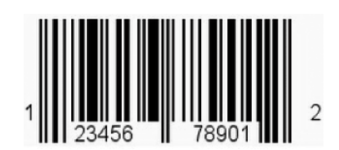


图 15-3 通过网页显示 UPC 条码

如表 15-1 所示是条码的参数及用途。

表 15-1 条码的 Web Service 参数和选项

参数	说明	参数选项
onlyimage	只显示图片或是包含选项	yes
encode	条码种类	UPC、EAN-8、EAN-13、UPC-E、S205、I2O5、I25、POSTNET、CODABAR、CODE128、CODE39 及 CODE93
bdata	条码号码	十进制数字
height	图片尺寸的高度	十进制数字
scale	图片缩放比例	十进制数字
bgcolor	后台图片颜色	十六进制,RGB 范围为 00～FF
color	条码颜色	十六进制,RGB 范围为 00～FF

STEP3：编写程序

通过 Python 程序语言获取文件并显示，需要使用 SimpleCV 和 urllib 函数库来编写程序，打开文本编辑器，输入以下程序。

```
$ sudo nano displayUPC.py
```

样例程序：ch09\ displayUPC.py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
3.  from SimpleCV import Image,Display
4.  from time import sleep
5.  import urllib
6.  url = "http://www.powenko.com/webservice/barcode/?encode = UPC - A&bdata = 120000000789&height = 50&scale = 2&bgcolor = %23FFFFff&color = %23000000&onlyimage = yes&file = &type = png&Genrate = Submit"
7.  filename = "out.jpg"
8.  urllib.urlretrieve(url,filename)       # 获取网络图片,并存在文件中
9.  Display1 = Display()
10. img = Image(filename)
11. img.save(Display1)  # 显示在屏幕画面上
12. while not Display1.isDone():
13.     sleep(1)
```

程序解说

第 8 行：获取网络图片，并存在文件中。

第 10 行：读入图片文件。

第 11 行：显示屏幕画面上。

5. 结果

执行 Python 程序，可以输入以下命令。

```
$ python displayUPC.py
```

正常执行时，会在屏幕上看到如图 15-4 所示的执行效果。



图 15-4　执行效果

6. 教学影片

完整的教学视频可以看 RaspberryPi_UPC_1_displayUPC. mov。

注意 如果要显示 QRCode,在 Google Chart 中有支持类似的功能,例如显示 Hello World,就可以通过以下网址获取图片。

```
https://chart.googleapis.com/chart?chs = 150x150&cht = qr&chl = Hello % 20world&choe =
UTF - 8
```

所以可以把本案例的样例程序修改一下,支持 QRCode 的产生,如图 15-5 所示。

请试着编写,完成后的程序可以看样例 displayQRCode. py,执行效果如图 15-6 所示。



图 15-5 通过 Google Chart 产生 QRCode



图 15-6 样例 displayQRCode. py 执行结果

15.2 条码扫描器

案例 85

1. 介绍

各位到大卖场购物结账时,常常会看到收银员拿着一个条码机扫描货品的编号并结算购物费用,所以有可能将树莓派变成一个商品扫描器吗? 可以通过 USB 摄像机获取影像输入,然后影像识别条码,并把条码的内容显示在屏幕上。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 1 个树莓派支持的 USB 的网络摄像机。

需要有以下软件:

ZBar

3. 步骤

先执行案例 84,再继续以下的操作。

STEP1:硬件准备

请将 USB 摄像机接在树莓派的硬件上,并重新开机使驱动程序执行,如图 15-7 所示。



图 15-7 硬件设备实际情况

STEP2:安装 ZBar 软件

需要安装相关的软件,请输入以下命令。

```
$ sudo apt - get install zbar - tools
$ sudo apt - get install qrencode
```

ZBar 是一套 Linux 环境上的扫描软件,如图 15-8 所示,可以支持和判断几乎市面上常见的条码编码,例如 EAN-13、UPC-A、UPC-E、EAN-8、Code 128、Code 39、Interleaved 2 of 5 和 QR Code 等。

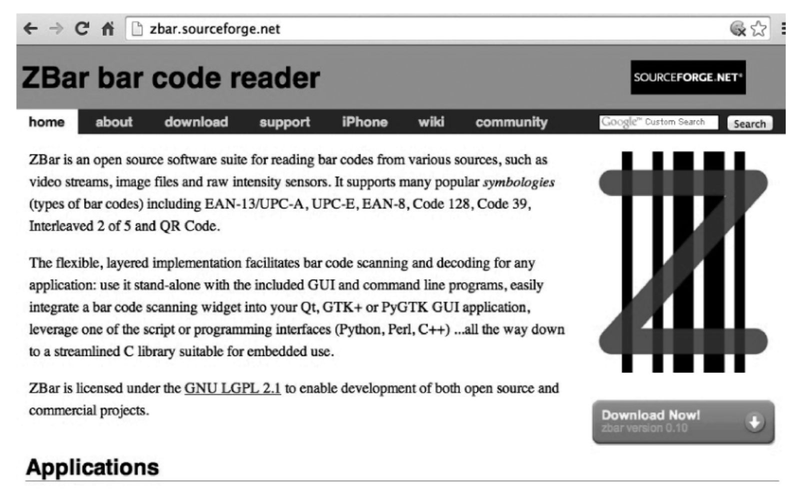


图 15-8 ZBar 的官方网站

STEP3：测试 ZBar 软件

为了测试软件是否可以正常工作，可以执行以下命令。

```
$ zbarcam --raw --prescale=320x240 /dev/video0
```

按照案例 84 介绍的条码实验，或者使用实际的条码，把 USB 摄像机对准到条码上，就会出现识别出来的数字。

4. 结果

只要把条码放到树莓派的 USB 镜头前面，就会自动识别条码并显示出条码的内容，执行效果如图 15-9 所示。



图 15-9　识别出来的条码数字

ZBar 软件对 Code 39 的识别能力最好，推荐先测试这个编码，再测试其他的条码。

5. 教学影片

完整的教学视频可以看 RaspberryPi_UPC_2_zbarscan-T。

15.3　扫描条码和存储到 CVS 数据库

案例 86

1. 介绍

本案例将进一步通过 USB 摄像机进行影像输入，然后使用 Python 程序语言和 SimpleCV 函数库来做影像显示，并通过 ZBar 影像识别条码的内容，把取得的数据存储到文件中，使用简易的数据库文件 CVS 格式。这个程序可以应用在很多地方，例如库存的盘点，商品的销售记录等。而且 CVS 文件格式可以把数据输出到其他数据库系统，例如微软的 Excel 或 Access，并进一步转换成 SQL。

2. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 1个树莓派支持的USB网络摄像机。

需要有以下软件:

(1) Python;

(2) SimpleCV函数库;

(3) python-zbar函数库。

3. 步骤

先执行案例85,再继续以下的操作。

STEP1:安装python-zbar函数库

安装Python的ZBar函数库,输入以下命令,如图15-10所示。

```
$ sudo apt-get install python-zbar
```

```
pi@raspberrypi ~/code/UPC $ sudo apt-get install python-zbar
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  python-zbar
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 41.0 kB of archives.
After this operation, 144 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main python-zbar armhf
 0.10+doc-8 [41.0 kB]
```

图15-10 安装Python的Zbar函数库

STEP2:编写程序

打开文本编辑器,输入以下程序,如图15-11所示。

```
$ sudo nano scanBarCode.py
```

样例程序:sample\ch09\ scanBarCode.py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
3.  print __doc__
4.  import time
5.  import csv
6.  from SimpleCV import Color,ColorCurve,Camera,Image,pg,np,cv
7.  from SimpleCV.Display import Display
8.  cam = Camera()
9.  display = Display((800,600))
10. data = "None"
11. mydict = dict()
12. myfile = "barcode-list.csv"
13. while display.isNotDone():
14.     display.checkEvents()
15.     img = cam.getImage()
16.     x1 = img.width/4
17.     x2 = img.height/4
18.     img.drawRectangle(x1,x2,img.width/2,img.height/2,color=Color.RED,width=3)
```

```
19.        if display.mouseLeft:                    #当用户按下左键
20.          img.drawText("reading barcode... wait",10,10)
21.          img.save(display)
22.          barcode = img.findBarcode()            #识别条码
23.          if barcode:
24.              print barcode
25.              data = str(barcode[0].data)        #取得条码数据
26.              print data
27.              if mydict.has_key(data):
28.                  mydict[data] = mydict[data] + 1
29.              else:
30.                  mydict[data] = 1
31.
32.          elif display.mouseRight:               #当用户按下右键
33.              print "save to CVS"
34.              target = open( myfile,"wb" )
35.              wtr = csv.writer( target )         #存储数据
36.              wtr.writerow(["item","count"])
37.              for d in mydict.items():
38.                  wtr.writerow(d)                #存储数据
39.              target.close()
40.        img.drawText("Click to scan.",10,10,color = Color.RED)
41.        myItem = "Last item scanned: " + data
42.        img.drawText(myItem,10,30)               #显示文字
43.        img.save(display)                        #画面显示
44.
```

```
powenko — pi@raspberrypi: ~/code/UPC — ssh — 80×24
  GNU nano 2.2.6            File: scanBarCode.py                    Modified

myfile = "barcode-list.csv"
while display.isNotDone():
    display.checkEvents()#check for mouse clicks
    img = cam.getImage()
$olor=Color.RED,width=3)
    if display.mouseLeft: # click the mouse to read
        img.drawText("reading barcode... wait",10,10)
        img.save(display)
        barcode = img.findBarcode()
        if barcode: # if we have a barcode
          print barcode
          data = str(barcode[0].data)
          print data
          if mydict.has_key(data):
             mydict[data] = mydict[data] + 1
          else:
             mydict[data] = 1
    elif display.mouseRight:
      #write to a CSV file.

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 15-11　通过文本编辑器编写程序

程序解说

第 13 行：通过无穷循环一直处理摄像机的影像。

第 15 行：获取摄像机的画面。

第 19～22 行：如果用户单击鼠标左键，开始做条码的识别操作。

第 23～30 行：如果取得条码，把条码的数据存储在数组中。

第 32～39 行：当用户单击鼠标右键，把所有的数据存储到文件中。

第 40～42 行：显示文字。

第 43 行：显示屏幕画面上。

程序编写完毕后，通过键盘的 Ctrl+O 组合键存储，Ctrl+X 组合键离开 nano 文本编辑器。

4. 结果

把条码放到树莓派的 USB 镜头前面，单击鼠标左键，就会自动扫描条码，并显示出条码的内容，如果要存储数据，单击鼠标右键，就会将数据存储到文件中，执行效果如图 15-12 所示，储存的内容如图 15-13 所示。



图 15-12　识别出来的条码数字

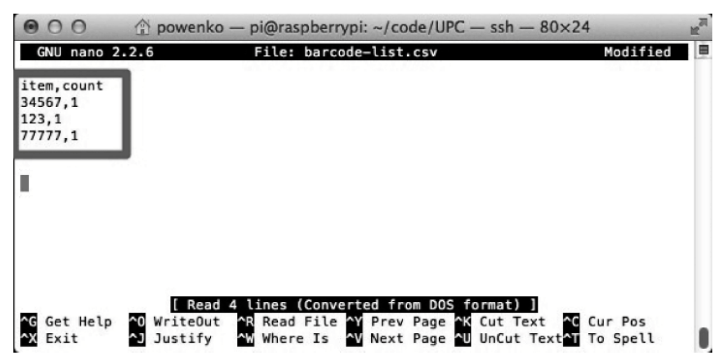


图 15-13　存储在 CSV 中的文件数据

5. 教学影片

完整的教学视频可以看 RaspberryPi_UPC_3_zbarPython-T。

第16章 P2P文件下载机Torrent

16.1 树莓派文件下载机 Torrent

案例 87

1. 介绍

本案例介绍文件共享软件 Torrent Deluge,如图 16-1 所示,很多人都会在个人计算机上开个文件共享软件,但很容易有信息安全上的顾虑,例如被骇客攻击和病毒袭击等,如果将树莓派变成一个独立的 Torrent 文件下载机器,就可以把重要的数据放在树莓派上,而且树莓派 Linux 系统的软件病毒也比 Windows 系统的少。



图 16-1　Deluge 软件官方网站

2. 目的

在树莓派安装和设置 Torrent,并且顺利地分享与下载文件。

3. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络连接设备;

(3) USB 外接硬盘。

4. 接线

确认网络功能正常。

把 USB 硬盘接到树莓派上。

5. 步骤

首先确认 USB 外接硬盘,已经按照本书的方法设置成功,并且可以正常运作,再来确认网络的设置正确,网络连接推荐用 RJ-45 接口连接,传输速度会比 WiFi 还快。

STEP1:更新安装程序 apt-get

请输入以下命令,获取最新的安装程序。

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

STEP2:安装 Deluge

输入以下命令安装 Deluge,如图 16-2 所示。

```
$ sudo apt-get install deluged deluge-console
```

```
pi@raspberrypi ~ $ sudo apt-get install deluged deluge-console
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
```

图 16-2 安装 Deluge

STEP3:产生 Deluge 的配置文件

产生 Deluge 的配置文件,执行它之后再关掉,这样才会产生配置文件,请输入以下命令执行,并把产生的配置文件备份为 auth.old,以避免改坏还有机会修复,如图 16-3 所示。

```
$ deluged
$ sudo pkill deluged
$ cp ~/.config/deluge/auth ~/.config/deluge/auth.old
```

```
$ deluged
$ sudo pkill deluged
$ cp ~/.config/deluge/auth  ~/.config/deluge/auth.old
```

图 16-3 产生 Deluge 的配置文件

STEP4:修改 auth 文件

使用 nano 文本编辑器,修改~/.config/deluge/auth。

```
$ sudo nano ~/.config/deluge/auth
```

在原本的文件结尾加上“账号:密码:10”,如图 16-4 所示,例如 pi:raspberry:10。

```
  GNU nano 2.2.6            File: /home/pi/.config/deluge/auth                Modif

localclient:e5c28542a2d6dd7d8ad9a151e638e0a7cb0ad89e:10
pi:raspberry:10

^G Get Help   ^O WriteOut   ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

图 16-4 修改 auth 文件

STEP5：引导 deluged

输入以下命令引导 deluged,如图 16-5 所示。

```
$ deluged
$ deluge-console
```

```
pi@raspberrypi ~ $ deluged
pi@raspberrypi ~ $ deluge-console
```

图 16-5 引导 deluged

STEP6：设置远程控制

进入 deluge-console 文本窗口,但为了以后操作便利,输入以下命令设置远程连接控制,如图 16-6 所示。

```
config 0s allow_remote True
config allow_remote
exit
```

```
Deluge 1.3.3 Console - localclient@127.0.0.1:58846

>>> config -s allow_remote True
Setting allow_remote to True..
Configuration value successfully updated.
* ConfigValueChanged: allow_remote: True
>>> config allow_remote
  allow_remote: True
C: 0 (200) D: 0.0 KiB U: 0.0 KiB DHT: 159
exit
```

图 16-6 设置远程连接控制

这样就已经完成文件共享软件 Deluge 的设置,可以进入 deluge-console 通过命令下载 Torrent 文件,如果有对命令兴趣的可以到官方网站上查看命令的教学。案例 88 会介绍图形化的工具,使用上就会容易得多。

6. 教学影片

完整的教学视频可以看 raspberryPi_torrent_1_deluge。

注意 提醒一下 Torrent 树莓派下载文件机的一个重点,因为这台设置的是一个 P2P 的文件共享机器,所以不只提供文件下载,同时文件也会上传分享给其他人,如果不想浪费上传网络流量,可以设置网络传输速度,或者下载成功后,就删除或迁移到其他地方。

16.2 管理树莓派下载机 Torrent

16.2.1 Windows 系统管理

案例 88

1. 目的

在浏览器中输入 http://dev.deluge-torrent.org/wiki/Download,下载对应的 Deluge,用来在 Windows 系统管理 P2P 机器。

2. 步骤

STEP1:下载和安装 Deluge

在浏览器中输入 http://dev.deluge-torrent.org/wiki/Download,下载 Windows 版本的 Deluge,如图 16-7 所示。

下载完毕后,选中进行安装,如图 16-8 所示。

STEP2:第一次执行时的设置

下载和安装 Deluge 完毕后,桌面上应该有 Deluge 图标,单击后执行 Deluge,第一次引导时,Windows 操作系统会弹出防火墙的警告消息,请单击 Unblock 按钮,如图 16-9 所示。

STEP3:找回 Connection Manager 连接管理器窗口

接下来就可以进入 Deluge 的窗口中,首先连接到树莓派下载机,需要有 Connection Manager 窗口,但有趣的是 Deluge 把它隐藏起来了,所以请通过以下操作打开 Connection Manager 窗口,如图 16-10 所示。

(1) 单击右上角的 Preference 按钮。

(2) Preference 窗口中,在左侧导航栏选择 Interface。

(3) 在 Interface 的 Classic Mode 模式中,关闭 Enable,如图 16-10 所示。

(4) 单击 OK 按钮离开 Preference 窗口。

(5) 把整个程序关闭,重新引导 Deluge。

图 16-7　下载 Windows 版本的 Deluge

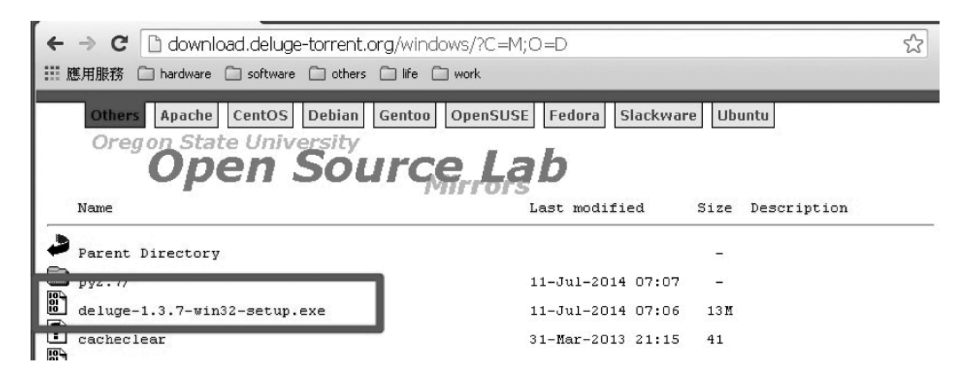


图 16-8　下载 Deluge

STEP4：设置 Connection Manager 连接管理器窗口

继续上一步操作，引导时就会看到 Connection Manager 连接管理器窗口，在工具栏上也会显示 Connection Manager 连接管理器窗口图标。请按照以下步骤完成连接到树莓派下载机的设置，如图 16-11 所示。



图 16-9　防火墙的警告消息

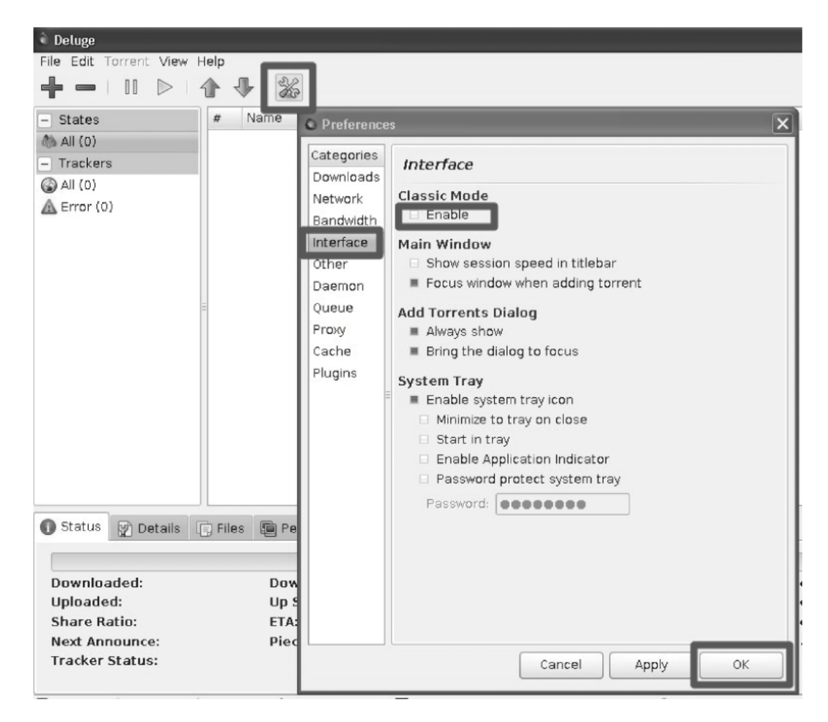


图 16-10　关闭 Classic Mode

（1）单击 Add 按钮。

（2）在 Add Host 窗口输入以下数据：

① Hostname：正确输入树莓派的网络地址。

② Port：网络端口号，请不要修改。

③ Username：用户名称，输入在～/. config/deluge/auth 文件中的用户名称。

④ Password 密码，输入在～/. config/deluge/auth 文件中的用户密码。

（3）在 Add Host 窗口，单击 Add 按钮，回到 Connection Manager 连接管理器窗口。

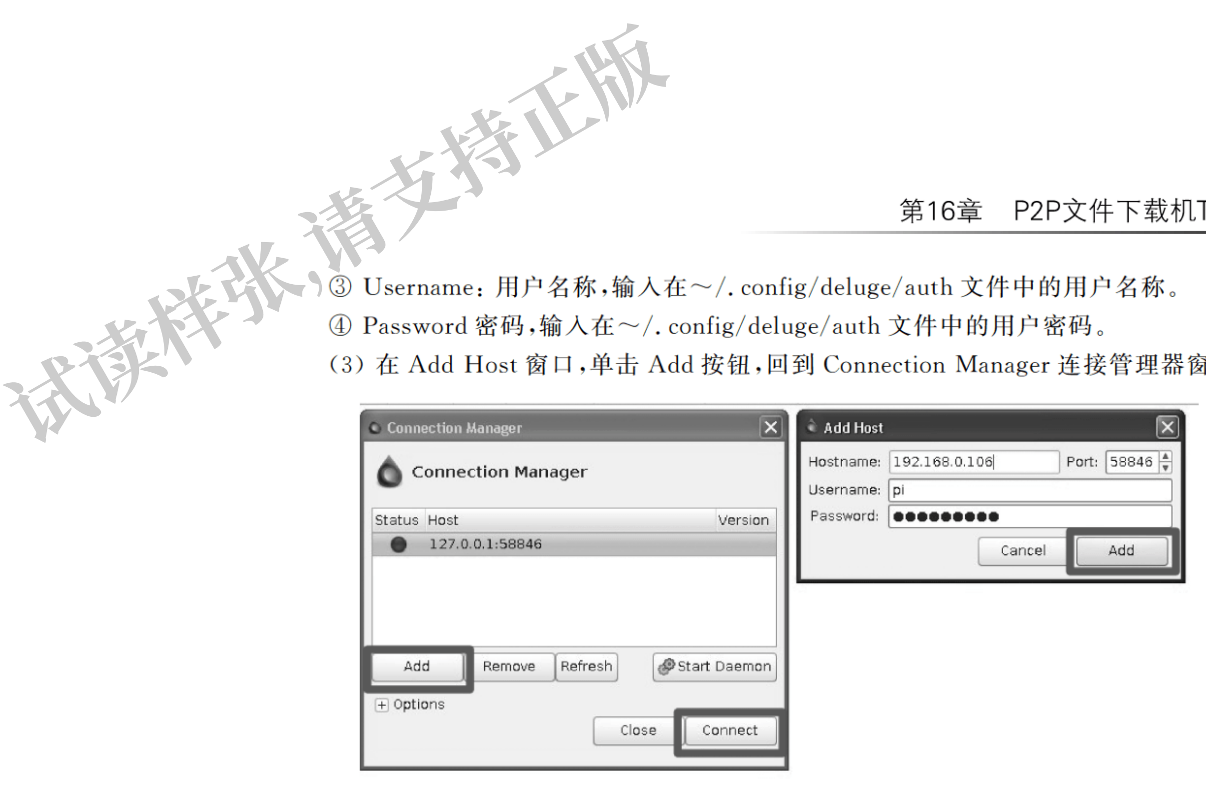


图 16-11　设置 Connection Manager 连接管理器窗口

STEP5：连接到树莓派下载机

如果 Deluge 找到树莓派下载机，就会显示绿色图标，选中该 Host，单击 Connect 按钮就可完成连接，如图 16-12 所示。



图 16-12　连接到树莓派下载机

STEP6：设置树莓派下载机文件存储路径

第一次使用时，选中 Preferences 设置窗口左侧导航栏中的 Downloads，设置树莓派下载机文件存储路径。

如图 16-13 所示的窗口中，指定到树莓派的外接硬盘位置上，按照外接硬盘的路径，分别修改：

（1）Download to：下载保存路径。

（2）Move completed to：完成下载后，复制保存路径。

(3) auto add . torrents from：自动添加. torrents 文件的路径。

(4) copy of . torrents to：复制. torrents 文件的路径。



图 16-13　设置树莓派下载机文件存储路径

3. 结果

设置完成，如图 16-14 所示，可以使用同一台计算机，用浏览器在网络上搜索和下载 P2P 的种子，也就是文件名后缀为. torrents 的文件，选中和下载. torrents 文件，就会在 Deluge 中自动执行并且下载，好处是个人计算机可以随时关机，树莓派下载机只要有电和网络，便会在后台模式下完整的下载文件，并且把文件存储在接在树莓派的外接硬盘中。

4. 教学影片

完整的教学视频可以看 raspberryPi_torrent_2_deluge_windows。

16.2.2　Mac 系统管理

案例 89

1. 目的

在 Mac 系统管理 P2P 机器。

2. 步骤

STEP1：下载和安装 Deluge

在 Mac 系统的机器上，打开浏览器连接到 http://dev. deluge-torrent. org/wiki/Download，下载 Mac 版本的 Deluge. app，如图 16-15 所示。



图 16-14　执行结果



图 16-15　下载 Mac 版本的 Deluge

下载完毕后，解压缩 Deluge. app....3. x86. tbz2 文件，产生 Deluge. app，把 Deluge. app 文件拖动到 Applications 进行安装，如图 16-16 所示。

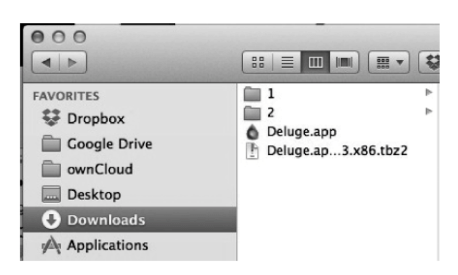


图 16-16　下载 Deluge

STEP2：第一次执行时的设置

Applications 的路径上会有 Deluge. app 图标，单击后执行 Deluge. app。第一次引导时，Mac 操作系统会弹出安全警告消息，请单击 Allow 按钮，如图 16-17 所示。

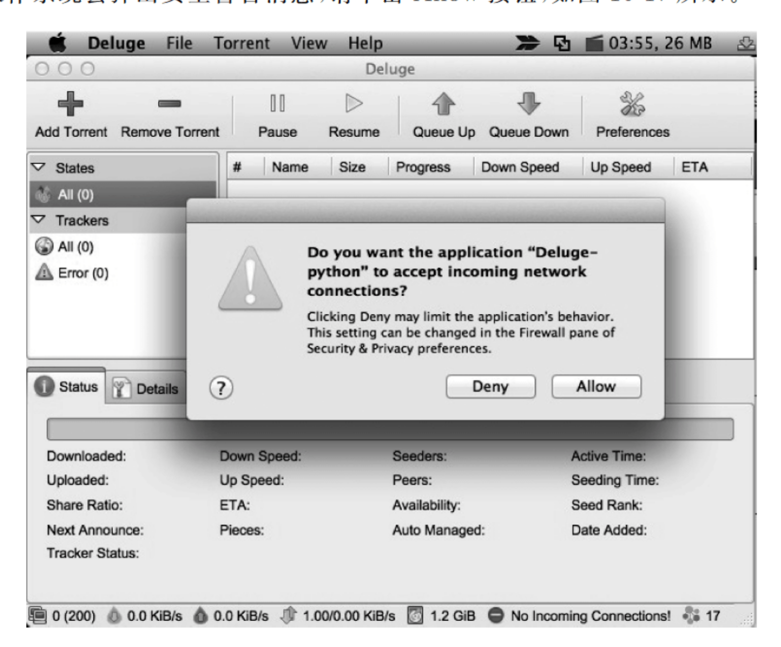


图 16-17　安全警告消息

STEP3：找回 Connection Manager 连接管理器窗口

同样 Connection Manager 连接管理器窗口也隐藏起来了，所以请通过以下操作打开 Connection Manager 窗口，如图 16-18 所示。

(1) 单击右上角的 Preference 按钮。

(2) Preference 窗口中，在左侧导航栏选择 Interface。

(3) 在 Interface 的 Classic Mode 模式中,关闭 Enable,如图 16-18 所示。

(4) 单击 OK 按钮离开 Preference 窗口。

(5) 把整个程序关闭,重新引导 Deluge。



图 16-18 关闭 Classic Mode

STEP4:设置 Connection Manager 连接管理器窗口

当继续上一步操作,引导时就会看到 Connection Manager 连接管理器窗口,在工具栏上也会显示 Connection Manager 连接管理器窗口图标。请按照以下步骤完成连接到树莓派下载机的设置,如图 16-19 所示。

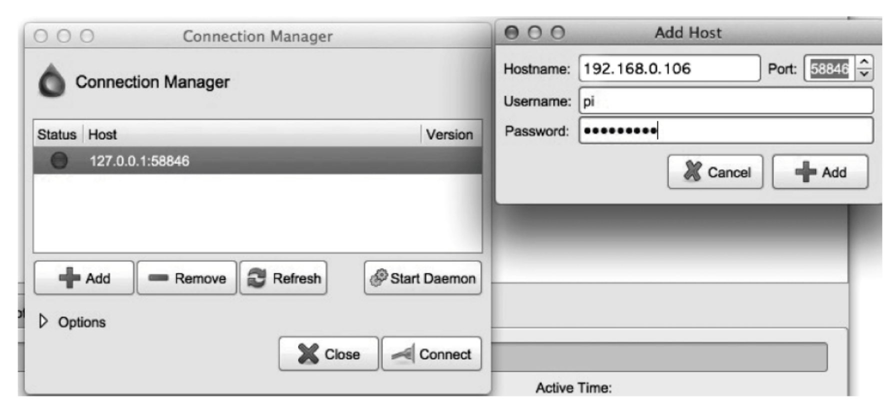


图 16-19 设置 Connection Manager 连接管理器窗口

(1) 单击 Add 按钮。

(2) 在 Add Host 窗口输入以下数据:

① Hostname:正确输入树莓派的网络地址。

② Port:网络端口号,请不要修改。

③ Username:用户名称,输入在~/.config/deluge/auth 文件中的用户名称。

④ Password:密码,输入在~/.config/deluge/auth 文件中的用户密码。

(3) 在 Add Host 窗口,单击 Add 按钮回到 Connection Manager 连接管理器窗口。

STEP5:连接到树莓派下载机

如果 Deluge 找到树莓派下载机,就会显示绿色图标,选中该 Host,单击 Connect 按钮就可完成连接,如图 16-20 所示。

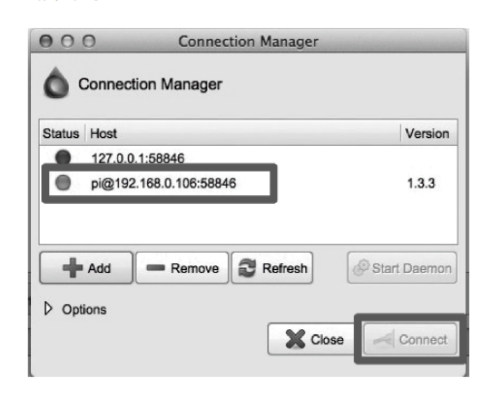


图 16-20 连接到树莓派下载机

STEP6:设置树莓派下载机文件存储路径

第一次使用时,选中 Preferences 设置窗口左侧导航栏中的 Downloads,设置树莓派下载机文件存储路径。

如图 16-21 所示的窗口中,指定到树莓派的外接硬盘位置上,按照外接硬盘的路径,分别修改:

(1) Download to:下载保存路径。

(2) Move completed to:完成下载后,复制保存路径。

(3) auto add .torrents from:自动添加.torrents 文件的路径。

(4) copy of .torrents to:复制.torrents 文件的路径。

3. 结果

设置结束后,可以在 Mac 计算机,用浏览器在网络上搜索和下载 P2P 的种子,也就是文件名后缀为.torrents 的文件,选中和下载.torrents 文件,就能引导 Deluge 自动执行并且下载,完成后的文件会存储在树莓派的外接硬盘上,如图 16-22 所示。



图 16-21 设置树莓派下载机文件存储路径

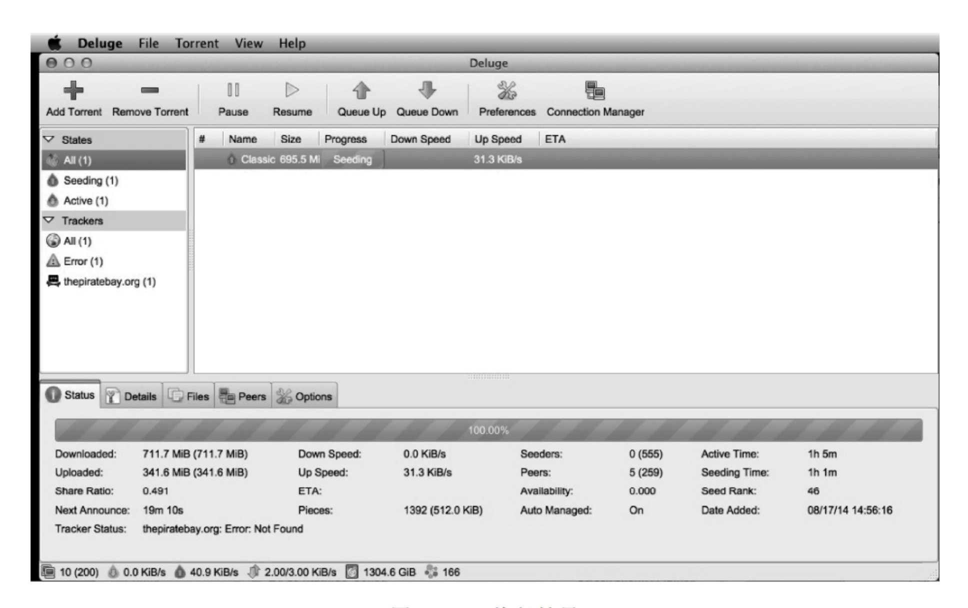


图 16-22 执行结果

16.2.3 Raspberry Pi 管理

案例 90

1. 目的

在树莓派系统管理 P2P 机器。

2. 步骤

STEP1：下载和安装 Deluge

按照以下指令，下载树莓派的版本的 Deluge，如图 16-23 所示。

```
$ sudo apt - get install deluge
```

```
pi@raspberrypi /media/usbhdd/torrent $ sudo apt-get install  deluge
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  deluge-gtk libcanberra-gtk3-0 libcanberra-gtk3-module libcanberra0 notification-daemon
  python-cairo python-glade2 python-gobject-2 python-gtk2 python-notify
```

图 16-23　下载树莓派版本的 Deluge

STEP2：进入窗口模式

输入以下指令，进入窗口模式。

```
$ startx
```

STEP3：引导 Deluge

依次单击 insternet→deluge 引导 Deluge，如图 16-24 所示。

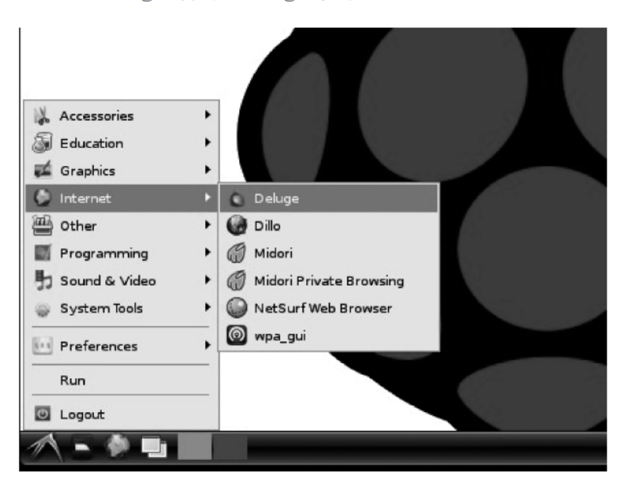


图 16-24　引导 Deluge

STEP4：找回 Connection Manager 连接管理器窗口

同样 Connection Manager 连接管理器窗口隐藏起来了，所以请通过以下操作打开 Connection Manager 窗口，如图 16-25 所示。

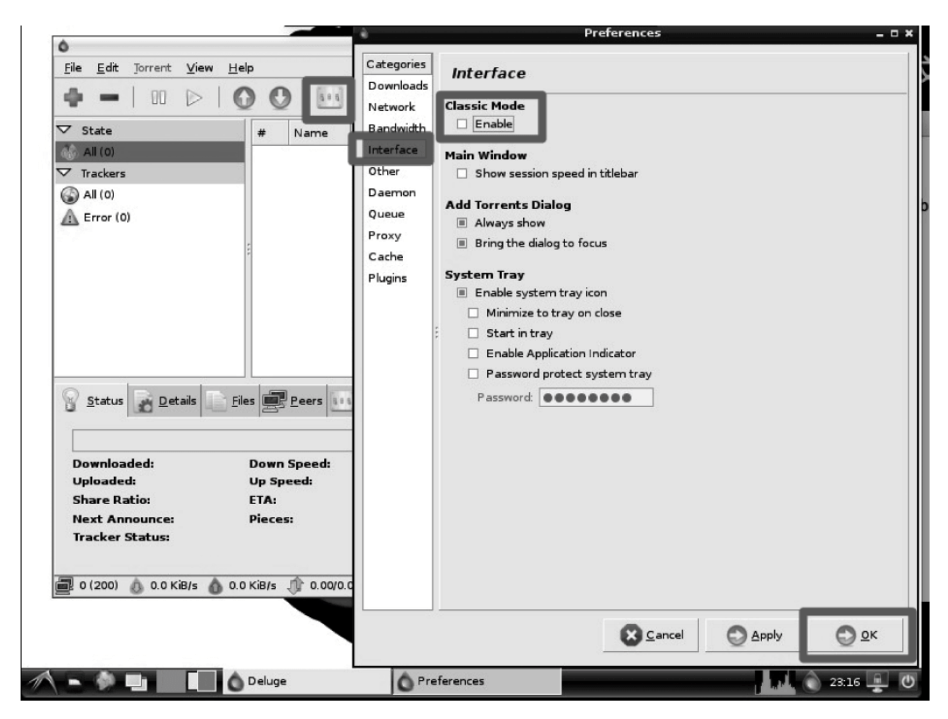


图 16-25　关闭 Classic Mode

(1) 单击右上角的 Preference 按钮。

(2) Preference 窗口中，在左侧导航栏选择 Interface。

(3) 在 Interface 的 Classic Mode 模式中，关闭 Enable，如图 16-25 所示。

(4) 单击 OK 按钮离开 Preference 窗口。

(5) 把整个程序关闭，重新引导 Deluge。

STEP5：设置 Connection Manager 连接管理器窗口

继续上一步操作，引导时就会看到 Connection Manager 连接管理器窗口，在工具栏上也会显示 Connection Manager 连接管理器窗口图标。请按照以下步骤完成连接到树莓派下载机的设置，如图 16-26 所示。

(1) 单击 Add 按钮。

(2) 在 Add Host 窗口输入以下数据：

① Hostname：正确输入树莓派的网络地址。

② Port：网络端口号，请不要修改。

③ Username：用户名称，输入在～/.config/deluge/auth 文件中的用户名称。

④ Password：密码，请输入在～/.config/deluge/auth 文件中的用户密码。

(3) 在 Add Host 窗口，单击 Add 按钮，回到 Connection Manager 连接管理器窗口。

注意 如果用同一台树莓派机器做下载和管理，这里的网络位置可以使用 IP Address 特别的网址，即指定同一台机器的网络地址 127.0.0.1。

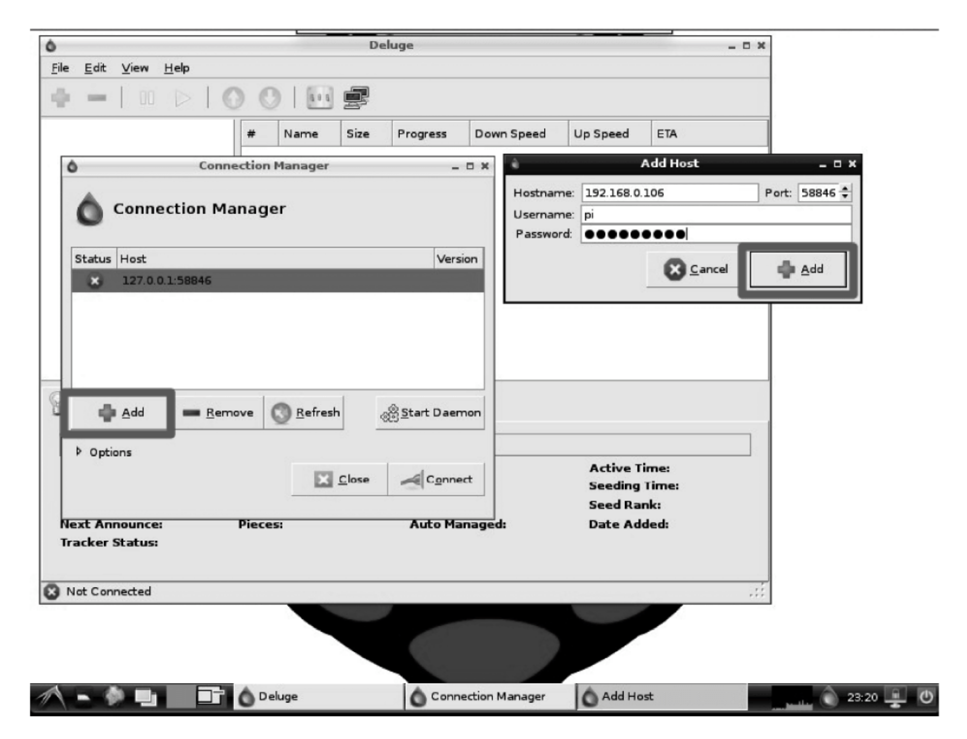


图 16-26 设置 Connection Manager 连接管理器窗口

STEP6：连接到树莓派下载机

如果 Deluge 找到树莓派下载机，就会显示绿色图标，选中该 Host，单击 Connect 按钮就可完成连接，如图 16-27 所示。

STEP7：设置树莓派下载机文件存储路径

选中 Preferences 设置窗口左侧导航栏中的 Downloads，设置树莓派下载机文件存储路径。

如图 16-28 所示的窗口中，指定到树莓派的外接硬盘位置上，按照外接硬盘的路径，分别修改：

(1) Download to：下载保存路径。



图 16-27　连接到树莓派下载机

(2) Move completed to：完成下载后，复制保存路径。

(3) auto add . torrents from：自动添加. torrents 文件的路径。

(4) copy of . torrents to：复制. torrents 文件的路径。

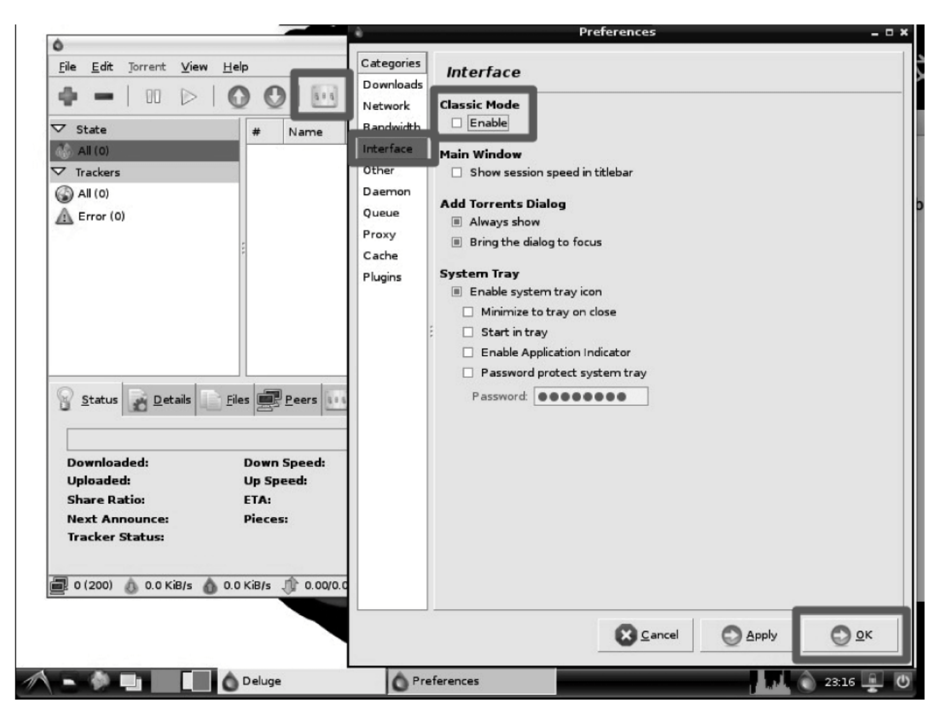


图 16-28　设置树莓派下载机存储文件路径

3. 结果

设置完成就可以在树莓派的窗口环境中,通过.torrents 文件下载文件,如图 16-29 所示。

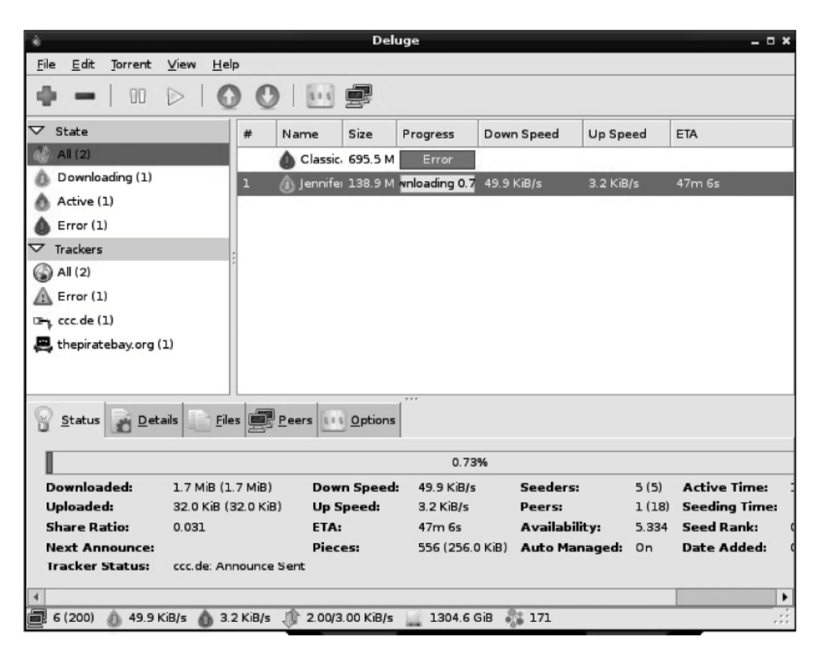


图 16-29 执行结果

4. 教学影片

完整的教学视频可以看 raspberryPi_torrent_2_deluge_raspberryPi。

第 17 章 自制 GameBoy 掌上型游戏机

17.1 树莓派游乐器主机——RetroPie

案例 91

1. 介绍

小时候非常喜欢玩任天堂的红白机和 GameBoy,如图 17-1 所示,当时为了购买游戏,努力帮忙家里打扫或打工,存下的钱去买勇者斗恶龙系列、龙战士等游戏,并且开始了没日没夜地打怪过关的日子,不知道你是否怀念当时那些游戏,本案例希望各位可以通过树莓派来重新怀念那些经典游戏。



图 17-1 Nintendo 任天堂红白机

案例核心在 RetroPie,这个惊人的仿真器软件,可以让树莓派转换成多种游乐器主机,当前可以支持如下硬件:

(1) Atari 2600;

(2) Game Boy Advance;

(3) Game Boy Color;
(4) Game Gear;
(5) MAME;
(6) NeoGeo;
(7) Sega Master System;
(8) Sega Megadrive;
(9) Nintendo 任天堂红白机;
(10) Super Nintendo 超级任天堂 SFC;
(11) PlayStation;
(12) n64;
等等。

2. 目的

在树莓派安装和设置游乐器主机的仿真器,并且可以顺利地执行游戏。

3. 设备

需要有以下硬件:
(1) 树莓派开发板;
(2) 网络连接设备。
需要有以下软件:
RetroArch

4. 接线

确认网络连接功能正常。

5. 步骤

STEP1:更新安装程序 apt-get
输入以下命令获取最新的安装程序。

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install -y git dialog
```

STEP2:安装声卡驱动程序
安装树莓派的声卡驱动程序,输入以下命令就可以顺利进行安装。

```
$ sudo apt-get install alsa-utils
$ sudo modprobe snd_bcm2835
$ sudo apt-get install libasound2-dev git-core
```

如果想测试声卡是否正常工作,可以把喇叭或耳机接到树莓派上,输入以下命令听是否播出声音。请自行把音乐文件的路径,按照实际状况进行调整。

```
$ sudo aplay /usr/share/xxx.wav
```

STEP3：下载安装相关的程序

安装相关的程序和工具，如图 17-2 所示。

```
$ mkdir ~/retro
$ cd retro
$ git clone git://github.com/libretro/RetroArch.git
$ cd RetroArch
```

```
pi@raspberrypi ~/game $ git clone  git://github.com/libretro/RetroArch.git
Cloning into 'RetroArch'...
remote: Counting objects: 80949, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 80949 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (80949/80949), 51.38 MiB | 2.32 MiB/s, done.
Resolving deltas: 100% (57164/57164), done.
pi@raspberrypi ~/game $ ls
RetroArch
pi@raspberrypi ~/game $ cd RetroArch/
pi@raspberrypi ~/game/RetroArch $ 
```

图 17-2　下载相关的函数库

通过 git clone 下载 Retro Arch 的源程序，如图 17-3 所示。官方网站为 https://github.com/libretro/RetroArch。

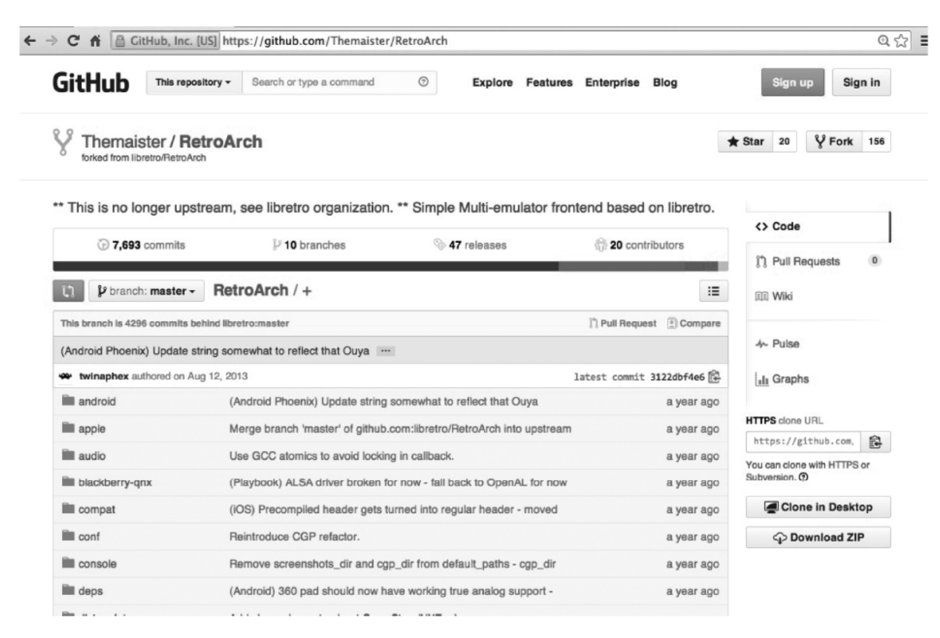


图 17-3　RetroArch 的源程序和官方网站

STEP4：设置和编译 RetroArch

输入以下命令来设置和编译 RetroArch 程序，如图 17-4 所示。

```
$ ./configure
$ make
```

编译需要不少时间,请耐心等待。

```
pi@raspberrypi ~/game/RetroArch $ ./configure
Checking operating system ... Linux
Checking for suitable working C compiler ... /usr/bin/gcc
Checking for suitable working C++ compiler ... /usr/bin/g++
Checking for availability of switch -std=gnu99 in /usr/bin/gcc ... yes
Checking for availability of switch -Wno-unused-result in /usr/bin/gcc ... yes
Checking for availability of switch -Wno-unused-variable in /usr/bin/gcc ... yes
```

图 17-4　设置和编译 RetroArch

执行以下步骤来下载和设置编译源程序,如图 17-5 所示。

```
$ git clone https://github.com/Themaister/RetroArch.git
$ cd RetroArch
$ ./configure
```

```
pi@raspberrypi ~/game/pocketsnes-libretro/RetroArch $ ./configure
Checking operating system ... Linux
Checking for suitable working C compiler ... /usr/bin/gcc
Checking for suitable working C++ compiler ... /usr/bin/g++
Checking for availability of switch -std=gnu99 in /usr/bin/gcc ... yes
Checking for availability of switch -Wno-unused-result in /usr/bin/gcc ... yes
Checking for availability of switch -Wno-unused-variable in /usr/bin/gcc ... yes
Checking function bcm_host_init in -lbcm_host ... yes
Checking for pkg-config ... /usr/bin/pkg-config
Checking presence of package egl ... no
Checking existence of -lEGL -lGLESv2 -lbcm_host -lvcos -lvchiq_arm ... yes
```

图 17-5　下载和设置编译

STEP5:下载 PicoDrive

因为树莓派的执行速度不够快,需要通过硬件解码来增加画面的流畅度,所以需要安装 PicoDrive,获取 PicoDrive 源程序的方法如下,如图 17-6 所示。

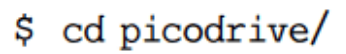
```
$ cd ..
$ ls
$ mkdir picodrive
$ cd picodrive
$ git clone https://github.com/libretro/picodrive.git
```

```
pi@raspberrypi ~/game/RetroArch $ cd ..
pi@raspberrypi ~/game $ ls
RetroArch
pi@raspberrypi ~/game $ mkdir picodrive
pi@raspberrypi ~/game $ cd picodrive
pi@raspberrypi ~/game/picodrive $ git clone https://github.com/libretro/picodrive.git
Cloning into 'picodrive'...
remote: Counting objects: 10564, done.
remote: Compressing objects: 100% (5126/5126), done.
remote: Total 10564 (delta 5363), reused 10564 (delta 5363)
Receiving objects: 100% (10564/10564), 9.89 MiB | 1.32 MiB/s, done.
Resolving deltas: 100% (5363/5363), done.
```

图 17-6　下载 PicoDrive

STEP6:调整 source code

下载完毕进入 picodrive/,如图 17-7 所示。



```
$ cd picodrive/
$ git submodule init && git submodule update
```

```
pi@raspberrypi ~/game/picodrive $ cd picodrive/
pi@raspberrypi ~/game/picodrive/picodrive $ git submodule init && git submodule update
Submodule 'cpu/cyclone' (git://notaz.gp2x.de/~notaz/cyclone68000.git) registered for path 'cpu/cyclone
'
Submodule 'platform/libpicofe' (git://notaz.gp2x.de/~notaz/libpicofe.git) registered for path 'platfor
m/libpicofe'
```

图 17-7　调整程序

STEP7：安装 libpng-dev 和 libsdl1.2-dev

安装所需要的 libpng-dev 和 libsdl1.2-dev 函数库。

```
$ sudo apt-get install libpng-dev
$ sudo apt-get install libsdl1.2-dev
```

STEP8：设置

通过以下命令设置 PicoDrive，如图 17-8 所示。

```
$ ./configure
```

```
pi@raspberrypi ~/game/picodrive/picodrive $ ./configure
architecture        arm
platform            generic
sound drivers        oss alsa sdl
C compiler          gcc
C compiler flags     -mfpu=vfp -I/usr/include/SDL -D_GNU_SOURCE=1 -D_REENTRANT -Wno-unused-result
libraries           -L/usr/lib/arm-linux-gnueabihf -lSDL -lasound -lpng  -lm
linker flags
libavcodec (mp3)    no
```

图 17-8　设置 PicoDrive

STEP9：编译 PicoDrive

完成整个设置之后，需要把 PicoDrive 编译支持 ARMv6E 的平台，输入以下命令进行编译，如图 17-9 所示。

```
$ make -f Makefile.libretro platform=armv6e
```

```
pi@raspberrypi ~/game/picodrive/picodrive $ make -f Makefile.libretro platform=armv6e
cc  -marm -DNO_ZLIB -fPIC -Wall -ggdb -falign-functions=2 -I. -DINLINE=inline -O2 -DNDEBUG -ffunction-
sections -D_SVP_DRC -DEMU_C68K -D_USE_DRZ80 -DDRC_SH2   -c -o platform/libretro/libretro.o platform/li
bretro/libretro.c
platform/libretro/libretro.c:902:13: warning: 'check_system_specs' defined but not used [-Wunused-func
tion]
cc  -marm -DNO_ZLIB -fPIC -Wall -ggdb -falign-functions=2 -I. -DINLINE=inline -O2 -DNDEBUG -ffunction-
sections -D_SVP_DRC -DEMU_C68K -D_USE_DRZ80 -DDRC_SH2   -c -o platform/common/mp3.o platform/common/mp
3.c
```

图 17-9　编译 PicoDrive

STEP10：复制 picodrive_libretro.so

把产生出来的 picodrive_libretro.so 复制到 RetroArch 的路径下，按照实际情况调整路径，如图 17-10 所示。

```
$ cp picodrive/picodrive/picodrive_libretro.so ~/game/ RetroArch
```

```
pi@raspberrypi ~/game/picodrive/picodrive $ ls
AUTHORS     config.mak  cpu       Makefile.libretro      platform  tools
ChangeLog   configure   jni       pico                   README    unzip
config.log  COPYING     Makefile  picodrive_libretro.so  skin      zlib
pi@raspberrypi ~/game/picodrive/picodrive $ cd ..
pi@raspberrypi ~/game/picodrive $ ls
picodrive
pi@raspberrypi ~/game/picodrive $ cd ..
pi@raspberrypi ~/game $ ls
picodrive  RetroArch
pi@raspberrypi ~/game $ cp picodrive/picodrive/picodrive_libretro.so RetroArch/
```

图 17-10 复制 picodrive_libretro. so 到 RetroArch 的路径下

STEP11：下载与编译 PC-Engine 仿真器

输入以下命令下载与编译 PC-Engine 仿真器，如图 17-11 所示。

```
$ git clone https://github.com/petrockblog/mednafen - pce - libretro.git
$ cd makemednaden - pce - libretro
$ make
```

```
pi@raspberrypi ~/game $ mkdir pcengine
pi@raspberrypi ~/game $ cd pcengine/
pi@raspberrypi ~/game/pcengine $ git clone https://github.com/petrockblog/mednafen-pce-libretro.git
Cloning into 'mednafen-pce-libretro'...
remote: Counting objects: 3549, done.
remote: Total 3549 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3549/3549), 5.30 MiB | 1020 KiB/s, done.
Resolving deltas: 100% (1497/1497), done.
Checking out files: 100% (2036/2036), done.
pi@raspberrypi ~/game/pcengine $ ls
mednafen-pce-libretro
pi@raspberrypi ~/game/pcengine $ cd mednafen-pce-libretro/
pi@raspberrypi ~/game/pcengine/mednafen-pce-libretro $ make
g++ -c -o mednafen/libretro/libretro.o mednafen/libretro/libretro.cpp -O3 -g -ffast-math  -funroll-loo
ps -fsigned-char -I. -Imednafen -Imednafen/include -Imednafen/intl -Imednafen/hw_cpu -Imednafen/hw_mis
c -Imednafen/hw_sound -Imednafen/hw_video  -DLSB_FIRST -DHAVE_MKDIR -DSIZEOF_DOUBLE=8  -DMEDNAFEN_VERS
```

图 17-11 下载与编译 PC-Engine 仿真器

编译完成后，输入以下命令复制产生出来的. so 文件。RetroArch 中按照实际情况调整路径。

```
$ cp *.so ～/game/RetroArch/
```

STEP12：下载与编译 NES 仿真器

输入以下命令下载与编译 NES 仿真器，如图 17-12 所示。

```
$ cd ～/game
$ git clone git://github.com/libretro/fceu - next.git
$ cd fceu - next/
$ pushd fceumm - code
$ make - f Makefile.libretro
```

```
pi@raspberrypi ~/game $ cd fceu-next/
pi@raspberrypi ~/game/fceu-next $ pushd fceumm-code
~/game/fceu-next/fceumm-code ~/game/fceu-next
pi@raspberrypi ~/game/fceu-next/fceumm-code $ make -f Makefile.libretro
```

图 17-12 下载与编译 NES 仿真器

编译完成后，输入以下命令复制产生出来的.so文件。RetroArch中按照实际情况调整路径。

```
$ cp *.so ~/game/RetroArch/
```

STEP13：编译其他的仿真器

顺便分享一下，如果要处理其他的仿真器，也需要把对应函数库进行编译。

```
https://github.com/libretro/libretro-prboom(Doom WADs)
https://github.com/libretro/fceu-next(NES)
https://github.com/libretro/imame4all-libretro(Arcade)
https://github.com/libretro/gambatte-libretro(GameBoy)
https://github.com/ToadKing/pocketsnes-libretro(SNES)
```

6. 结果

完成之后，把树莓派接到屏幕上，输入以下命令，并且指向游戏的文件，就能够温经典游戏。为了尊重作者和知识产权，请大家支持原版。

```
$ cd ~/game/RetroArch/
```

如果要执行NES的游戏，请输入以下命令。

```
$ ./retroarch -L fceumm_libretro.so xxxxxx.nes
```

如果要执行Sega的游戏，请输入以下命令。

```
$ ./retroarch -L picodrive_libretro.so xxxxxx.md
```

执行影片可以观看视频raspberryPi_Game_3_demo，如图17-13所示。



图17-13　RetroArch的执行画面

7. 教学影片

完整的教学视频可以看raspberryPi_Game_1_T.mov和raspberryPi_Game_2_T.mov，这个案例的视频比较长，主要是花很多时间在编译源程序。

17.2 游戏摇杆

案例 92

1. 介绍

案例 91 中,各位顺利地拥有自己的树莓派游乐器主机,但实际使用键盘和鼠标玩游戏,总是觉得少了流畅的操作感,原来是少了摇杆这个主要的功能。所以在本案例中,加上摇杆的功能,介绍如何把 XBox 360 摇杆,修改并加入到树莓派中,当成游乐器主机的游戏摇杆,这样就可以用 XBox 360 摇杆来玩游戏,如图 17-14 所示。

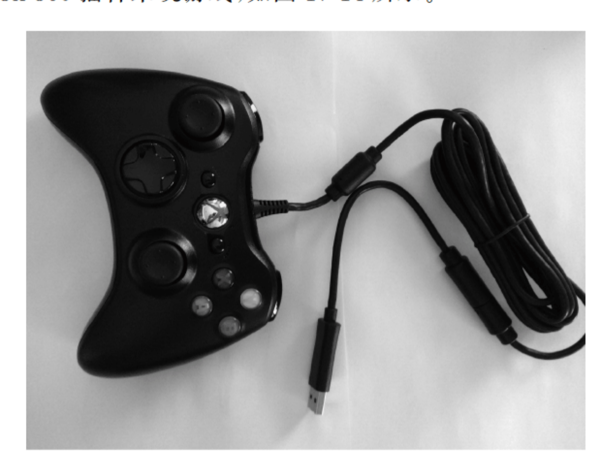


图 17-14 XBox 360 的 USB 摇杆

2. 目的

在树莓派上面安装和设置 XBox 360 遥控器,并且让游乐器主机的仿真器顺利地使用该遥控器。

3. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 网络连接设备;

(3) XBox 360 的 USB 摇杆。

需要有以下软件:

Retro Arch

4. 接线

确认网络连接功能正常。

5. 步骤

STEP1：更新安装程序 apt-get

输入以下命令获取最新的安装程序。

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install -y git dialog
```

STEP2：安装驱动程序

输入以下命令安装 XBox 360 的 USB 摇杆驱动程序，如图 17-15 所示。

```
$ sudo apt-get install xboxdrv
```

```
pi@raspberrypi ~ $ sudo apt-get install xboxdrv
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  xboxdrv
```

图 17-15　安装 XBox 360 的 USB 摇杆驱动程序

把 USB 的 XBox 360 的摇杆接到树莓派上之后，输入以下的命令重新开机。

```
$ sudo reboot
```

STEP3：确认 XBox 360 的 USB 摇杆安装驱动程序

重新开机后输入以下的命令，确认驱动程序是否正确安装，如图 17-16 所示。

```
$ lsusb
```

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 8301:0089
Bus 001 Device 005: ID 045e:028e Microsoft Corp. Xbox360 Controller
```

图 17-16　确认 Microsoft Corp. XBox360 Controller

STEP4：设置按键对应文件

输入以下命令重新设置摇杆的对应文件，首先把旧的删除。

```
$ sudo rmmod xpad
```

STEP5：创建新的按键对应文件

官方的样例文件如图 17-17 所示，有很多现成的样例可以参考，官方网址是 https://github.com/Grumbel/xboxdrv/tree/master/examples。

这里以任天堂的按键对应表来设置，如表 17-1 所示。

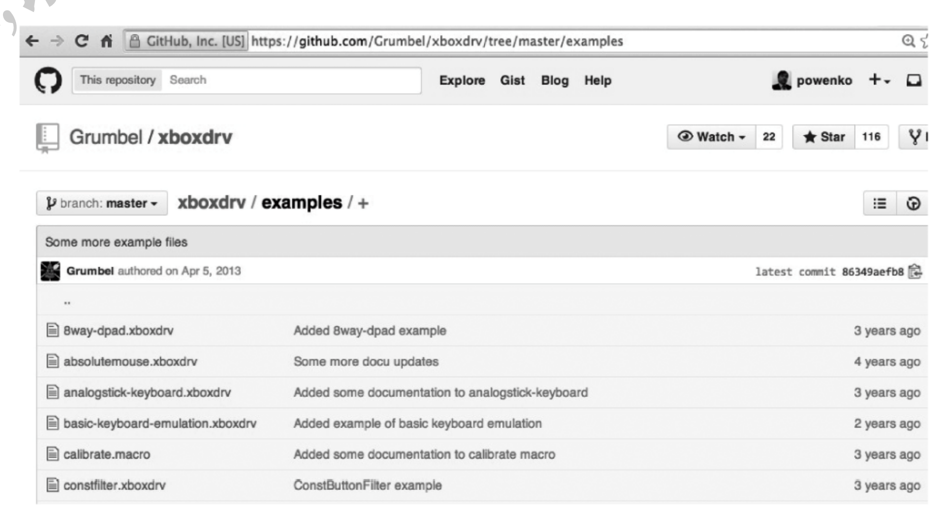


图 17-17　键盘对应的样例文件

表 17-1　键盘对应表

xboxdrv 定义键	任天堂仿真器	按键表 Keycode
du	上	KEY_UP
dd	下	KEY_DOWN
dl	左	KEY_LEFT
dr	右	KEY_RIGHT
guide	q 离开	KEY_Q
start	p 暂停	KEY_P
b	x 跳	KEY_X
a	z 攻击	KEY_Z

创建一个新的 xboxdrv 按键对应文件。

```
$ nano /home/pi/myconfig.xboxdrv
```

输入以下数据进行存储,这里有键盘的定义和处理鼠标的动作,也就是说,可以把 XBox 360 的 USB 摇杆当成鼠标使用,如图 17-18 所示。

样例程序:sample\ch16\myconfig. xboxdrv

```
[xboxdrv]
ui-clear=true
silent = true

[axis-sensitivity]
X2 = -0.5
```

```
Y2 = -0.5

[ui-axismap]
x2^dead:4000 = REL_X:750:-1
y2^dead:4000 = REL_Y:750:-1


y1^dead:6000^invert = rel-repeat:REL_WHEEL:1:50
x1^dead:6000 = rel-repeat:REL_HWHEEL:1:50

lt = KEY_VOLUMEDOWN:20
rt = KEY_VOLUMEUP:20

[ui-buttonmap]
a = KEY_Z
b = KEY_X
x = BTN_MIDDLE
y = KEY_ENTER

rb = BTN_LEFT
lb = BTN_RIGHT

tl = KEY_BACKSPACE
tr = KEY_SPACE

dl = KEY_LEFT
dr = KEY_RIGHT
du = KEY_UP
dd = KEY_DOWN

start = KEY_P
back   = KEY_BACK
guide = KEY_ESC

# EOF #
```

STEP6：编辑配置文件

通过文本编辑器设置/etc/rc.local 文件。

```
$ sudo nano /etc/rc.local
```

在 fi 和 exit 0 之间加上以下命令，就可以加上一个摇杆，如图 17-19 所示。

```
rmmod xpad
xboxdrv --trigger-as-button --id 0 --led 2 --deadzone 4000 --silent -c /home/pi/
myconfig.xboxdrv& sleep 1
```

```
GNU nano 2.2.6                    File: /home/pi/myconfig.xboxdrv

[xboxdrv]
ui-clear=true
silent = true

[axis-sensitivity]
X2=-0.5
Y2=-0.5

[ui-axismap]
x2^dead:4000 = REL_X:750:-1
y2^dead:4000 = REL_Y:750:-1

y1^dead:6000^invert = rel-repeat:REL_WHEEL:1:50
x1^dead:6000 = rel-repeat:REL_HWHEEL:1:50

lt = KEY_VOLUMEDOWN:20
rt = KEY_VOLUMEUP:20

[ui-buttonmap]
a = KEY_Z
b = KEY_X
x = BTN_MIDDLE
y = KEY_ENTER

rb = BTN_LEFT
lb = BTN_RIGHT

tl = KEY_BACKSPACE
tr = KEY_SPACE
                              [ Read 41 lines ]
^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

图 17-18　创建新的按键对应文件

```
GNU nano 2.2.6                    File: /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

rmmod xpad
xboxdrv --trigger-as-button --id 0 --led 2 --deadzone 4000 --silent -c /home/pi/myconfig.xboxdrv  & sleep 1
# xboxdrv --trigger-as-button --id 1 --led 3 --deadzone 4000 --silent & sleep 1


exit 0

                              [ Read 25 lines ]
^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

图 17-19　编辑/etc/rc.local 文件

存储文件后,离开文本编辑器。

如果树莓派上接上 2 个摇杆,那就修改 id 和 led 的号码。

```
xboxdrv --trigger-as-button --id 0 --led 2 --deadzone 4000 --silent & sleep 1
xboxdrv --trigger-as-button --id 1 --led 3 --deadzone 4000 --silent & sleep 1
```

接上 4 个摇杆,那就进行如下修改。

```
xboxdrv --trigger-as-button --id 0 --led 2 --deadzone 4000 --silent & sleep 1
xboxdrv --trigger-as-button --id 1 --led 3 --deadzone 4000 --silent & sleep 1
xboxdrv --trigger-as-button --id 2 --led 4 --deadzone 4000 --silent & sleep 1
xboxdrv --trigger-as-button --id 3 --led 5 --deadzone 4000 --silent & sleep 1
```

如果使用无线 XBox360 摇杆,那就把 id 修改为 wid。

```
xboxdrv --trigger-as-button --wid 0 --led 2 --deadzone 4000 --silent & sleep 1
```

并且针对每一个摇杆,通过参数-c,指到相对应的按键对应文件。

STEP7:重新开机

设置完毕后,重新打开树莓派就可以使用摇杆了。

```
$ sudo reboot
```

6. 结果

完成之后,只要引导游戏仿真器,就可以测试摇杆的设置是否正确,如图 17-20 所示。

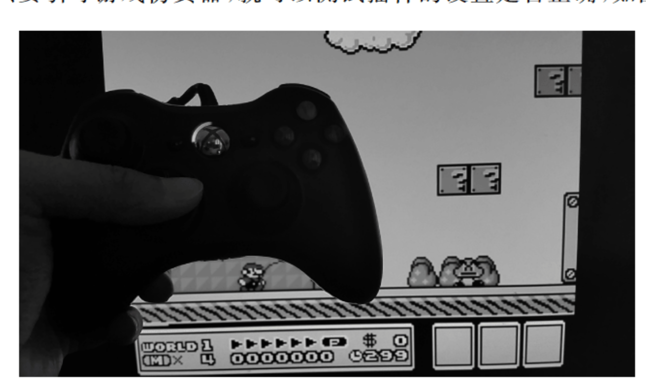


图 17-20　执行画面

因为在 xboxdrv 定义文件中有针对鼠标的操作,所以只要进入窗口环境,就可以把遥控器来当成鼠标使用,如图 17-21 所示。

7. 教学影片

完整的教学视频可以看 RaspberryPi_GameController.mov。

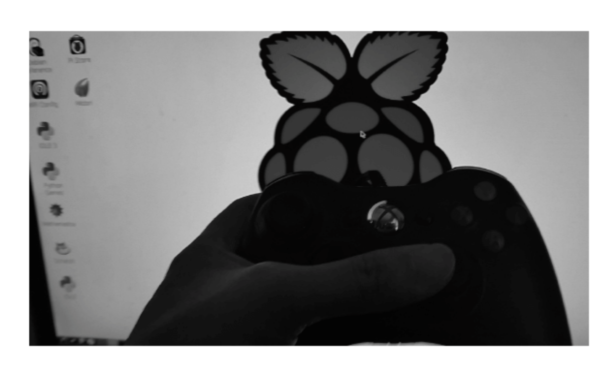


图 17-21　把遥控器当成鼠标的执行画面

17.3　自制 GPIO 键盘输入设备

案例 93

1. 目的

本案例介绍如何自制键盘。设计原理是通过树莓派上面的 GPIO,读入自制的 6 个按键的操作,当玩家按下按键时,系统程序便会检测到并仿真出键盘的操作,输出特定的字符。

本案例通过 pikeyd 开源代码来实现目的,官方网站如图 17-22 所示。

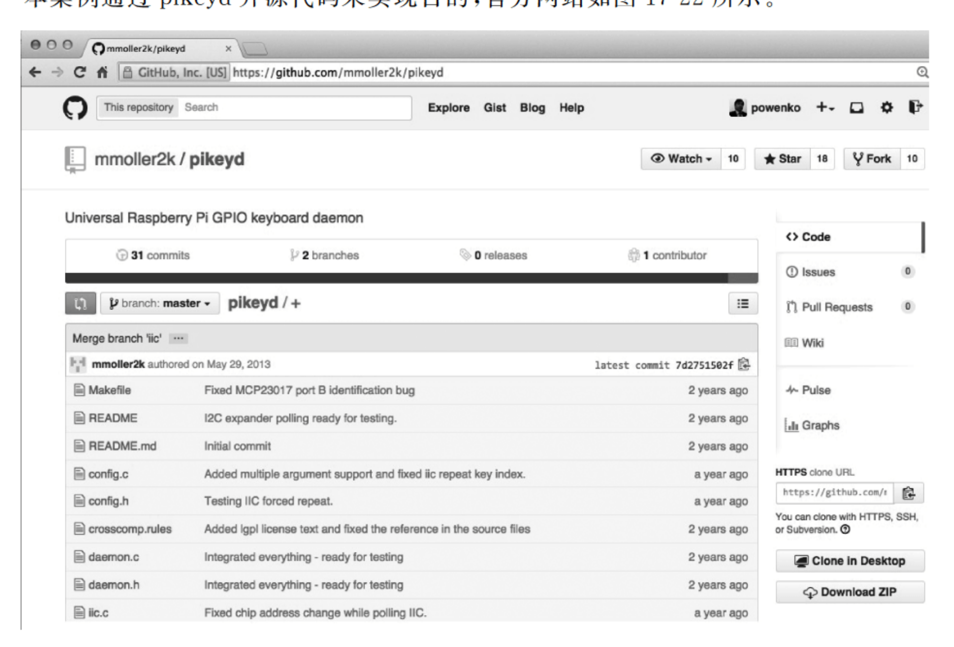


图 17-22　pikeyd 应用

2. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 6 个按钮；

(3) 6 个 220Ω 的电阻；

(4) 线材；

(5) 面包板。

3. 接线

按照以下硬件线路图分别把 6 个按钮接到树莓派的 GPIO 引脚上，硬件线路如图 17-23 所示，树莓派的 GPIO 引脚如图 7-24 所示。

功能	上	下	左	右	A 键	B 键
树莓派的引脚	GPIO 17 引脚 Pin 11	GPIO 23 引脚 Pin 16	GPIO 10 引脚 Pin 19	GPIO 9 引脚 Pin 21	GPIO 25 引脚 Pin 22	GPIO7 引脚 Pin 26
对应键盘码	KEY_UP	KEY_DOWN	KEY_LEFT	KEY_RIGHT	KEY_X	KEY_Z

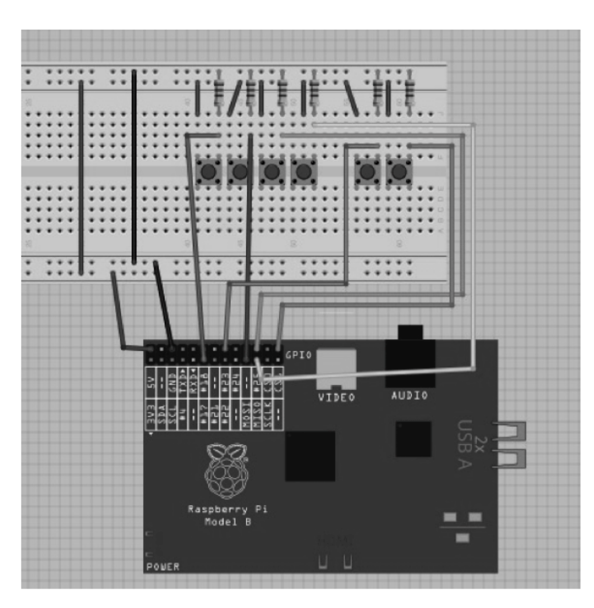


图 17-23　硬件线路

4. 步骤

STEP1：连接硬件线路

首先把树莓派关机，按照硬件线路图连接好硬件线路，确定线路正确后再开机，如图 17-25 所示。

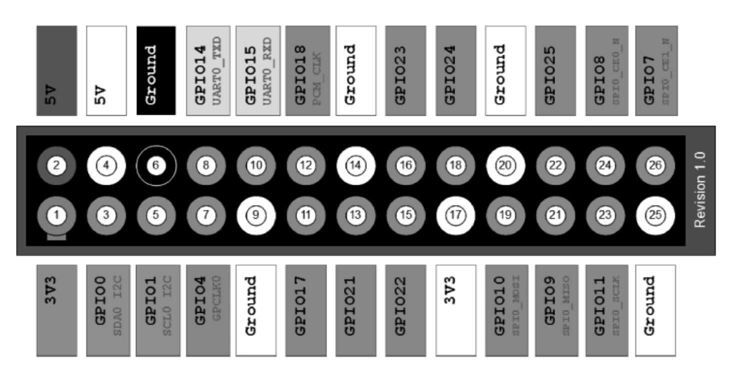


图 17-24 GPIO 引脚号码对应图

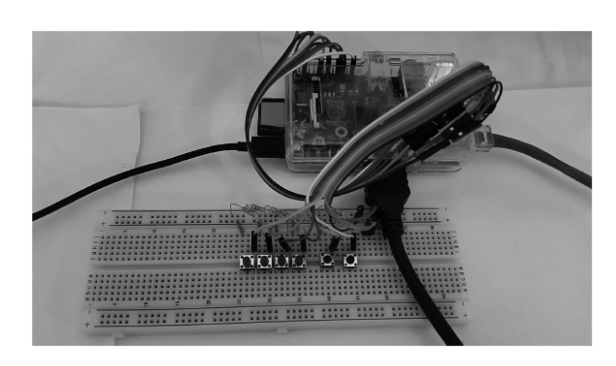


图 17-25 实际接好的硬件线路

STEP2：更新安装程序 apt-get

输入以下命令获取最新的安装程序。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

STEP3：取得 pikeyd 源程序

输入以下命令，先创建一个路径，再来获取最新的 pikeyd 源程序，如图 17-26 所示。

```
$ mkdir gpiokey
$ cd gpiokey
$ wget https://github.com/mmoller2k/pikeyd/archive/master.zip
```

```
pi@raspberrypi ~/gpiokey $ ls
master.zip  pikeyd-master
pi@raspberrypi ~/gpiokey $ cd pikeyd-master/
pi@raspberrypi ~/gpiokey/pikeyd-master $ make
arm-linux-gnueabihf-gcc -O2 -Wstrict-prototypes -Wmissing-prototypes -I/home/pi/
RPi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/arm-linux-gnueabih
f/include -I/home/pi/RPi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbi
```

图 17-26 获取 pikeyd 源程序

STEP4：解压缩

完成后输入以下命令解压缩，如图17-27所示。

```
$ unzip master.zip
```

STEP5：编辑器

获取的是C语言的源代码，输入以下命令编译源程序并产生执行文件，如图17-28所示。

```
$ cd pikeyd-master/
$ make
```

```
pi@raspberrypi ~/gpiokey $ unzip master.zip
Archive:  master.zip
7d2751502f94f785f7da2020087e981e6e686ae4
   creating: pikeyd-master/
  inflating: pikeyd-master/Makefile
  inflating: pikeyd-master/README
  inflating: pikeyd-master/README.md
```

图17-27　解压缩

```
pi@raspberrypi ~/gpiokey $ ls
master.zip  pikeyd-master
pi@raspberrypi ~/gpiokey $ cd pikeyd-master/
pi@raspberrypi ~/gpiokey/pikeyd-master $ make
```

图17-28　获取pikeyd源程序

STEP6：添加按键对应表

通过文本编辑器打开/etc/pikeyd.conf。

```
$ sudo nano /etc/pikeyd.conf
```

将以下内容写入，如图17-29所示。

```
KEY_UP 17
KEY_DOWN 23
KEY_LEFT 10
KEY_RIGHT 9
KEY_X 25
KEY_Z 7
```

完成后通过Ctrl+O组合键存储和Ctrl+X组合键离开文本编辑器。

```
  GNU nano 2.2.6              File: /etc/pikeyd.conf

KEY_UP 17
KEY_DOWN 23
KEY_LEFT 10
KEY_RIGHT 9
KEY_X 25
KEY_Z 7

                          [ Read 7 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图17-29　GPIO按下时转换为键盘的对应表

注意 这里的数字指树莓派上的 GPIO 的号码。

STEP7：测试

输入以下命令执行测试，如图 17-30 所示，当按下 A 键、B 键时，屏幕上会出现 X 和 Z 的文字，按下其他的 GPIO 按键，就会出现上下左右的键盘反应，如图 17-31 所示。

```
$ sudo ./pikeyd
```

```
pi@raspberrypi ~/gpiokey/pikeyd-master $ sudo ./pikeyd
/dev/i2c-1: No such file or directory
Config file is /etc/pikeyd.conf
Polling 6 GPIO pin(s).
Found 0 I/O expander(s).
Ready.
Joystick init OK.
Press ^C to exit.
```

图 17-30　执行 pikeyd 的反应

注意 这里的反应不会出现在 SSH 中，请用屏幕接上 HDMI 线才会有正确的反应。

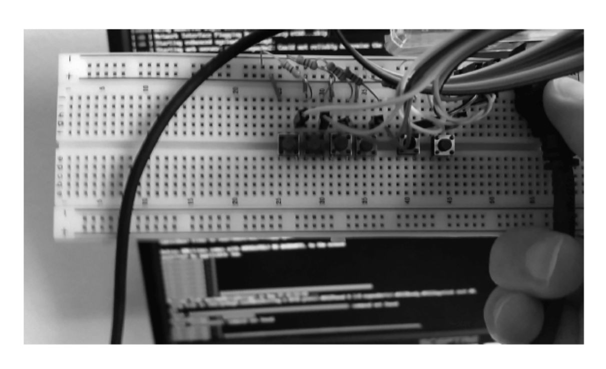


图 17-31　实际反应

STEP8：设置开机自动执行

为了下次开机可以拥有同样的效果，先输入以下命令，设置这个程序的执行权限。

```
sudo chmod + x /home/pi/gpiokey/pikeyd - master/pikeyd
```

打开文本编辑器。

```
$ sudo nano /etc/rc.local
```

在文本的 exit 0 之前加上以下文字，特别是参数-d 用来设置执行 daemon 就是守护程序，如图 17-32 所示。

```
sudo /home/pi/gpiokey/pikeyd - master/pikeyd  - d &
```



图 17-32　设置开机自动执行

5. 结果

完成之后,把树莓派重新开机打开游戏仿真器,就可以测试自制的 GPIO 摇杆是否正常工作,如图 17-33 所示。

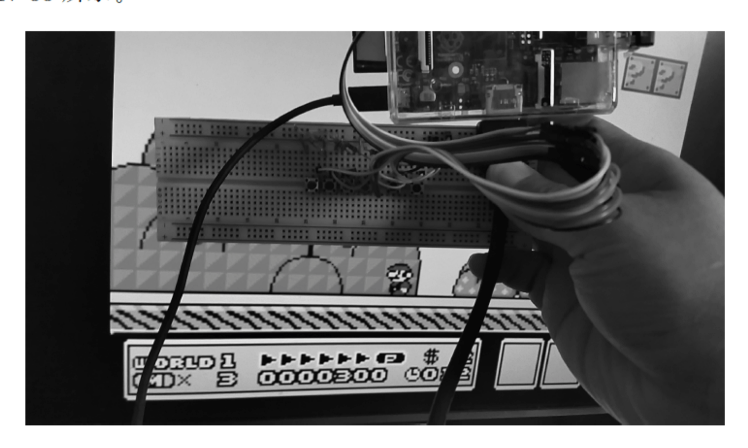


图 17-33　执行画面

提醒一下,这里使用 GPIO 产生键盘按键的反应,也可以用此方法做出输入数据的设备。

6. 教学影片

完整的教学视频可以看 RaspberryPi_Game_GPIOKEY.mov。

17.4 自制掌上型游乐器主机

案例 94

不知道各位手上是否有坏的游乐器主机,如果有就可以把里面的主机板拿出来,把树莓派放里面,网络上有很人就是这样处理,可以重温旧梦回忆起以前玩游乐器的点点滴滴。

如果没有这样的设备,也可以用纸张自行剪裁外壳,把树莓派放入自制的游乐器主机盒子里。如果手上有 3D 打印机,通过打印出模型自行组装,就会有更有趣的效果。

到 Adafruit 网站上,如图 17-34 所示,下载 GameBoy 游乐器主机盒子的 3D 打印文件 STL。网址为:https://learn.adafruit.com/pigrrl-raspberry-pi-gameboy?view=all。

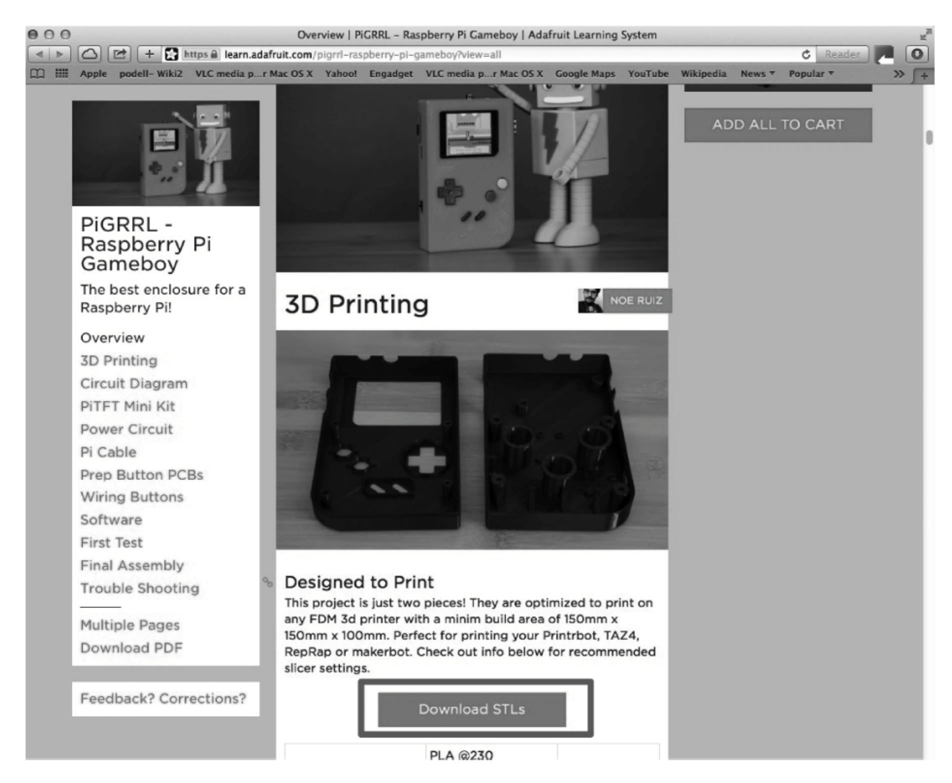


图 17-34 下载 GameBoy 的 3D 模型文件

还有一个 GameBoy 游乐器主机盒子的 3D 打印文件 STL,如图 17-35 所示。网址为:http://www.thingiverse.com/thing:321624/#files。

屏幕要如何处理呢?推荐使用 AV 端子的小尺寸屏幕来当作 GameBoy 的屏幕,例如如图 17-36 所示的 AV 端子的 2.5 寸 TFT 屏幕,注意供电的问题,树莓派的 GPIO 只有 3V 和



图 17-35　另一种 GameBoy 的 3D 模型文件

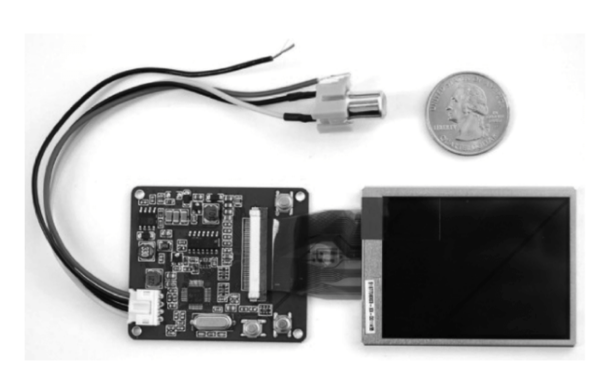


图 17-36　2.5 寸 TFT 屏幕

5V 的直流电,推荐不要超过 1A 的电流使用量。

供电使用手机 USB 移动电源,使用 2A 以上的 USB 充电器并且体积合适,这样才能放入盒子中,如图 17-37 所示。

按键输入请参考案例 93 自制按键,并且转换键盘对应游乐器主机的按键。把这些配备处理好放入盒子中,就是一个自制 GameBoy 的主机了,完成组装成品效果如图 17-38 所示。

另外一个造型的树莓派掌上型游乐器,完成组装成品如图 17-39 所示。



图 17-37 超规模移动电源



图 17-38 Adafruit Industries 自制的 GameBoy 成品



图 17-39 3Derp 自制的 GameBoy 成品

第18章 遥控器开关

18.1 使用树莓派显示红外线遥控器的信号

案例95

1. 介绍

本案例介绍如何自制一个遥控器。为了达到此目的,需要了解以下知识:

(1) 树莓派怎么接收遥控器的信号。

(2) 了解遥控器的原理。

首先介绍遥控器,家中可能有很多的遥控器,开门的遥控器、看电视的遥控器,现在连窗帘、冷气机都有遥控器,本案例介绍如何将树莓派通过遥控器做互相沟通和联络,这样的技术可以把玩家的操作和需求,在第一时间传递给树莓派,达到彼此之间的交互,并把比较复杂的计算交给树莓派进行计算机,做出合适的反应与处理操作。

遥控器是由发射器、接收器和微处理器组成。接收器和微处理器大部分都在家电主机上,例如电视机。遥控器产生不同的编码脉冲,以决定不同的按键,而遥控器输出各种以红外线为媒介的控制脉冲信号,当接收器收到特定的脉冲之后,将红外线信号进行放大、限幅、检波、整形后送到微处理器,微处理器根据不同的信号,再做不同按键的反应。注意,根据发射不同编码指令脉冲设计成相应的控制功能,这些功能是生产厂商预先设计好的,再刻录在遥控器里面,所以各种遥控器是不能通用的,而且各家厂商会刻意做不同的调整,以免互相干扰。

发送和接收端之间的联系有两种方法,分别是:

(1) IR(Infra Red)红外线;

(2) RF(Radio Frequency)射频。

想知道遥控器是哪一种类,可以用照相机的预览镜头对着遥控器,当按下时发射端看到红外线的亮度就是使用IR红外线,一般肉眼是看不到的。RF射频就看不到红外线灯光,这是因为RF射频遥控器是用频率,所以它的范围比较广,可以穿越阻碍物,不会妨碍到数据传输。

2. 目的

通过 IR Receiver 接收遥控器送出来的数据,并且显示在屏幕上。

3. 设备介绍

遥控器的接收端称为 Receiver,IR Receiver 的外观如图 18-1 所示。

4. 硬件规范

红外线接收器有 3 个引脚,如图 18-2 所示,这 3 个引脚分别会依照不同厂牌的红外线传感器,该 3 个引脚的功能都不太一样,所以请按照实际使用的硬件来调整接线,或者参考表 18-1 比较常见的红外线接收器引脚。



图 18-1 IR Receiver 的外观

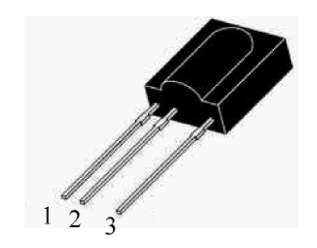


图 18-2 红外线接收器 PNA4602 的引脚

表 18-1 比较常见的红外线接收器引脚

红外线接收器编号	最左边引脚	中间引脚	最右边引脚
VS 1838B	接地	电源	信号输出
PNA4602	接地	电源	信号输出
TSOP4836/SFH5110	信号输出	接地	电源
TSOP1836/NJK61H380	信号输出	接地	电源
TSOP1736/SFH506/TFMS5360	接地	电源	信号输出
SFH505A	电源	接地	信号输出
IS1U60	信号输出	接地	电源
PIC120435	信号输出	接地	电源

IR 遥控器如图 18-3 所示,本案例只要确定是 IR 红外线遥控器就可以使用。如果不确定,可以把家中所有的遥控器都准备好,只要经过本案例测试,可以读出遥控器编码,那就是 IR 红外线遥控器。

5. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 面包板;

(3) 1 个 IR Receiver;



图 18-3　红外线遥控器的外观

(4) 1 个 IR 遥控器；

(5) 接线若干。

需要有以下软件：

(1) Python；

(2) Python GPIO 函数库。

6. 接线

参考表 18-2 和图 18-4，把红外线遥控器和接收器连接到树莓派的线路上。

表 18-2　引脚表

树莓派引脚	组件引脚
引脚 24 红外线接收器	红外线接收器信号输出引脚
5V(引脚 2)	红外线接收器电源引脚
GND (引脚 6)	红外线接收器接地引脚

再次提醒不同的 IR 接收器的引脚都不太一样，但都是接到树莓派的电源、接地和引脚 24。

7. 步骤

STEP1：更新安装软件

输入以下命令更新安装软件，如图 18-5 所示。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

STEP2：下载 Python 和 GPIO 安装相关的程序

安装相关的程序和工具。

```
$ sudo apt - get install python - dev
$ sudo apt - get install python - rpi.gpio
```

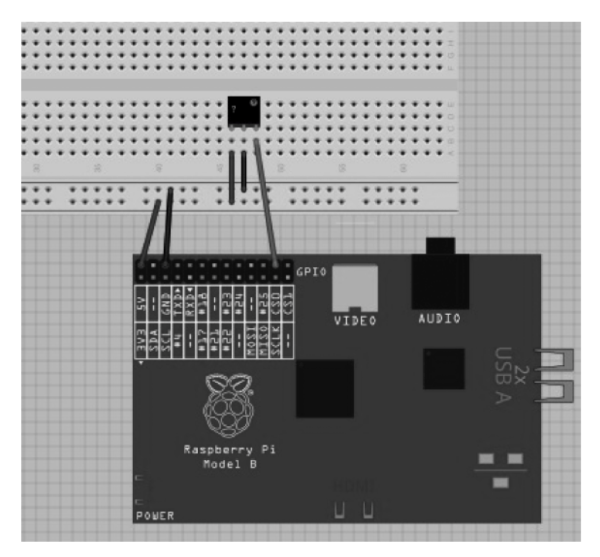


图 18-4　红外线遥控器和接收器连接到树莓派的线路上

```
pi@raspberrypi ~/code $ sudo apt-get update
Hit http://repository.wolfram.com stable Release.gpg
Get:1 http://raspberrypi.collabora.com wheezy Release.gpg [836 B]
Hit http://repository.wolfram.com stable Release
Get:2 http://archive.raspberrypi.org wheezy Release.gpg [490 B]
Get:3 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Get:4 http://raspberrypi.collabora.com wheezy Release [7,532 B]
Get:5 http://archive.raspberrypi.org wheezy Release [7,227 B]
Get:6 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Hit http://repository.wolfram.com stable/non-free armhf Packages
```

图 18-5　更新安装软件

STEP3：编写代码

编写 Python 程序语言用来显示遥控器按键按下去的信号。

```
$ sudo nano irread.py
```

样例程序：sample\ch20_IRread\irread.py

```
1.   #!/usr/bin/env python
2.   # author: Powen Ko
3.   import time
4.   import RPi.GPIO as GPIO
5.   GPIO.setmode(GPIO.BOARD)
6.   GPIO.setwarnings(False)                    # 不要显示警告消息
7.   GPIO_ECHO = 24
8.   print "IR Signal"
9.   GPIO.setup(GPIO_ECHO,GPIO.IN)              #设置引脚
10.
11.  start = 0
12.  stop = 0
13.  while True:
```

```
14.    start = time.time()
15.        duringDown = start - stop;
16.    while GPIO.input(GPIO_ECHO) == 0:
17.        start = time.time()
18.
19.    while GPIO.input(GPIO_ECHO) == 1:
20.        stop = time.time()
21.
22.    #计算波型的时间
23.    duringUp = stop - start
24.    info = "Up:%7.f" % (duringUp * 100000) + ",Down:%7.f" % (duringDown * 100000)
25.    print info
26.    if duringUp > 0.1:
27.        print("-- -- -- -- -- -- -- -- --")
28. # 离开并恢复引脚
29. GPIO.cleanup()
```

程序解说

第 1 行：执行环境的定义文档，Python 程序存放路径。

第 2 行：注释的写法，前面加个#就成为注释。

第 7～9 行：设置引脚 24 为输入引脚。

第 16～17 行：等待引脚变成高电位。

第 19～20 行：等待引脚变成低电位。

第 23 行：计算引脚的高电位的时间。

第 24 行：显示数据。

完成后按下 Ctrl+O 组合键存储，Ctrl+X 组合键离开文本编辑器 nano。

STEP4：执行和测试

其他硬件接到树莓派上并准备好遥控器，执行以下命令。

```
$ sudo python irread.py
```

把遥控器对准 IR 接收器，按下遥控器上的按钮，就会显示出信号了。

8. 结果

执行这个应用时，除了执行程序，还需要把遥控器对准红外线接收器，按下按键，如果画面上显示出一串的数字，那就是红外线接收器收到的遥控器按键的信号，如图 18-6 所示。建议先把其中几个遥控器按键记录下来，因为下一个案例会使用到。

9. 教学影片

完整的教学视频可以看 RaspberryPi_IR_Read_Signal。

注意 对 C 语言熟悉的开发者，可以使用一个处理遥控器相当普遍的函数库 LIRC，如图 18-7 所示，处理遥控器的按键反应和工作。

```
-------------------
Up:    450,Down:    161
Up:     27,Down:     83
Up:     38,Down:     77
Up:     27,Down:     85
Up:     26,Down:     88
Up:     35,Down:     77
Up:     30,Down:     84
Up:     26,Down:     84
Up:     35,Down:     79
Up:    168,Down:     56
Up:    163,Down:     65
Up:    166,Down:     52
Up:    173,Down:     54
Up:    170,Down:     58
Up:    165,Down:     55
Up:    139,Down:     87
Up:    167,Down:     52
Up:     49,Down:     59
Up:     49,Down:     52
Up:    168,Down:     59
Up:    165,Down:     59
Up:     43,Down:     55
Up:     46,Down:     56
Up:     54,Down:     53
Up:     52,Down:     49
Up:    168,Down:     60
Up:    167,Down:     54
Up:     47,Down:     58
Up:     51,Down:     52
Up:    166,Down:     49
Up:    164,Down:     60
Up:     98,Down:    128
Up:    122,Down:    106
Up:   3950,Down:    116
Up:    222,Down:     95
```

图 18-6 红外线接收器收到的信号



图 18-7 C 语言的遥控器函数库 LIRC

18.2 遥控器控制树莓派上的LED灯开关

案例96

1. 目的

继续案例95,把刚刚读到的遥控器信号,加以判断来确认玩家按下的是哪一个按键,来调整LED灯的亮度。本案例的挑战在于,一般的红外线遥控器即使是同一个按键,每次送出来的信号都会有误差,所以在开发程序时需要把这些问题解决掉。还有不同型号的遥控器,送出来的信号也都不一样,请按照实际的遥控器,把数据稍微做调整就可正常工作了。

2. 设备介绍

同案例95。

3. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 面包板;

(3) 1个IR Receiver;

(4) 1个IR遥控器;

(5) 2个LED灯;

(6) 接线若干。

需要有以下软件:

(1) Python;

(2) Python GPIO函数库。

4. 接线

参考表18-3和图18-8,把红外线遥控器和接收器连接到树莓派的线路上。

表18-3 引脚表

树莓派引脚	组 件 引 脚
引脚24	红外线接收器信号输出引脚
5V(引脚2)	红外线接收器电源引脚
GND(引脚6)	红外线接收器接地引脚,第一个LED灯的短脚,第二个LED灯的短脚
引脚23	第一个LED灯的长脚
引脚24	第二个LED灯的长脚

5. 步骤

本案例是由"18.1 使用树莓派显示红外线遥控器的信号"扩展过来,需首先完成此章节。

STEP1:整理出遥控器4个按键的信号

按照案例95把遥控器4个按键的信号整理出来,并通过Python的array方法排列。

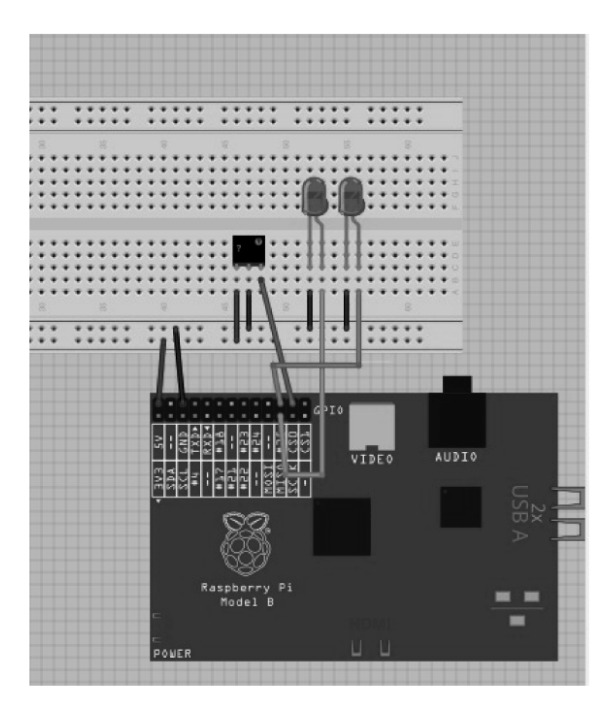


图 18-8　红外线遥控器和接收器连接到树莓派的线路上

STEP2：编写代码

打开 nano 文本编辑器输入以下代码。

```
$ sudo nano irled.py
```

样例程序：sample\ch20-IRLED\ir2led.py

```
1.   #!/usr/bin/env python
2.   # author: Powen Ko
3.   import time
4.   import RPi.GPIO as GPIO
5.   KeyCurrent = range(100)
6.   i = 0                                    # 遥控器按钮的信号
7.   key2 = [446,167,29,82,29,84,26,85,37,78,27,84,25,88,32,81,38,77,163,65,161,52,158,
     57,168,
8.   54,162,57,162,50,163,56,157,65,51,53,55,48,46,58,162,56,162,49,29,87,50,49,52,
     62,159,
9.   57,165,48,165,52,19,98,54,63,160,49,166,50,171,50,4001,61,221,69];
10.  key4 = [445,152,22,89,26,87,34,77,42,77,32,82,33,79,38,75,29,85,155,69,166,52,165,
     53,162,
11.  61,165,52,141,86,163,52,166,60,49,50,56,58,50,55,166,49,47,57,50,59,58,51,54,
     57,166,
```

```
12. 56,162,49,165,52,43,71,167,58,164,50,163,52,166,62]
13. key6 = [447,157,29,83,31,81,34,82,27,79,43,77,24,87,29,84,34,78,149,80,148,80,140,
    85,144,
14. 78,166,57,169,55,165,61,163,50,50,56,165,48,48,60,166,51,163,51,53,52,166,48,55,
    53,166,
15. 49,52,57,164,49,54,57,50,51,158,72,48,61,165,50,4003,56,223,63]
16. key8 = [443,161,29,82,34,78,30,85,32,83,36,76,22,91,24,87,37,78,167,53,164,57,165,
    50,164,
17. 61,161,63,163,51,168,52,166,51,50,63,169,50,46,50,53,55,161,49,57,58,165,53,53,
    52,166,
18. 56,52,56,164,59,168,51,49,58,166,49,44,57,167,51]
19.
20. keys = [key2,key4,key6,key8]
21. GPIO.setmode(GPIO.BOARD)
22. GPIO.setwarnings(False)
23. GPIO_ECHO  =  24
24. print "IR Signal"
25.
26. #函数——判断用户按下哪个按钮
27. def Funcheck(a,b,i):
28.     if (a - b)> 0.1:   #判断传送是否结束
29.         i = 0
30.
31.
32.     for j1 in range(len(keys)):
33.         isThisKey = True
34.         for j2 in range(5,len(keys[j1]) - 4):
35.             value = keys[j1][j2]
36.             currentValue = KeyCurrent[j2]
37.             rangeValue = 100                    # 注意：请按照实际情况调整此变量
38.             if ((currentValue - rangeValue)< value and (currentValue + rangeValue)> value)
    == False :
39.                 isThisKey = False
40.                 # print " error:j1 = %d" % j1 + " j2 = %d" % j2  + " currentValue =
    %d" % currentValue + " value = %d " % value
41.                 break
42.         if isThisKey == True:
43.             print "Get it key %d" % j1        #显示用户按下哪一个按钮
44.             if j1 == 0:
45.                 GPIO.output(21,True)
46.             elif j1 == 1:
47.                 GPIO.output(21,False)
48.             elif j1 == 2:
49.                 GPIO.output(22,True)
50.             elif j1 == 3:
51.                 GPIO.output(22,False)
```

```
52.             i = 0                       #清除记录
53.             for j3 in range(len(KeyCurrent)):
54.                 KeyCurrent[j3] = 0
55.
56.             return i
57.
58.
59.
60.  # 主程序——设置 GPIO
61.  GPIO.setup(GPIO_ECHO,GPIO.IN)
62.  GPIO.setup(21,GPIO.OUT)
63.  GPIO.setup(22,GPIO.OUT)
64.  start = 0
65.  stop = 0
66.    while True:
67.        start  =  time.time()
68.        duringDown = start - stop;
69.
70.        while GPIO.input(GPIO_ECHO) == 0:
71.            start  =  time.time()
72.
73.        while GPIO.input(GPIO_ECHO) == 1:
74.            stop  =  time.time()
75.            i = Funcheck(stop,start,i)
76.
77.         duringUp  =  stop - start
78.         KeyCurrent[i] = (duringUp  * 100000)
79.         i = i + 1
80.         KeyCurrent[i] = (duringDown * 100000)
81.         i = i + 1
82.         i = Funcheck(stop,start,i)
83.
84.  #结束离开程序并撤销 GPIO 的设置
85.  GPIO.cleanup()
86.
```

程序解说

第 1 行：执行环境的定义文档，Python 程序存放路径。

第 2 行：注释的写法，前面加个#就成为注释。

第 27～56 行：Funcheck 函数判断用户按下的按键是否为这 4 个按钮。

第 34 行：这个循环判断为何不是从第一个判断到最后一个呢？这是因为最前面和最后面的信号是等待用户按下按钮的时间。

第 44～57 行：判断出遥控器的按键，作出相应反应。

编写 Python 程序，一部分矩阵数据会按照实际遥控器的按键信号不同做些调整。

(1) 程序的第 7～8 行,需要一张实际的遥控器按键图,将检测出的信号加以调整。

(2) 程序的第 37 行,按照按键的数字,调整这个变量的尺寸,原则上讲变量越小越精准。

(3) 程序的第 28 行,这个变量的数据用来决定,等待多久时如果没有反应,就算是遥控器的信号结束。

完成后按下 Ctrl+O 组合键存储,Ctrl+X 组合键离开文本编辑器 nano。

STEP3:执行和测试

其他硬件接到树莓派上并准备好遥控器,执行以下命令。

```
$ sudo python ir2led.py
```

把遥控器对准 IR 接收器,按下遥控器上的按钮,就会显示出信号了。

6. 结果

执行这个应用时,在遥控器按下自定按钮,就会让 LED 灯做出开或关的反应,如图 18-9 所示。如果失败,可参考案例 95 来调整程序第 7～18 行中的输出数据。

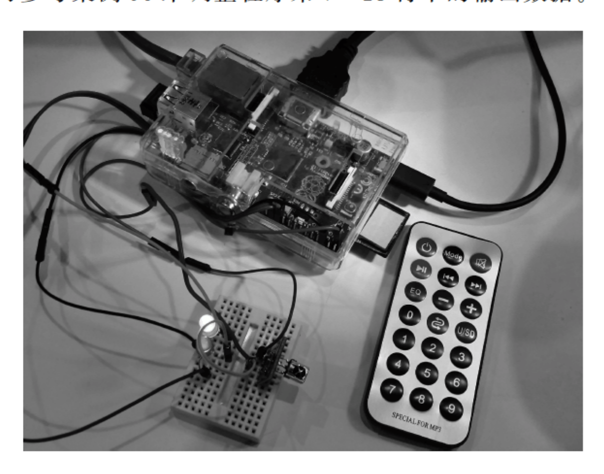


图 18-9　实际执行效果

7. 教学影片

完整的教学视频可以看 RaspberryPi_IR_CheckKeys 和 RaspberryPi_IR_CheckKeys_LEDOnOff。

第19章 遥控汽车应用

19.1 晶体管控制直流电动机

案例97

1. 介绍

直流电动机用的地方非常多,如图19-1所示,遥控汽车、电风扇都可看到直流电动机的应用,这里通过树莓派来控制电动机的开关。



图19-1 直流电动机

本案例还会用到晶体管,晶体管有两种常用的功用:

(1) 放大信号(电流放大):主要用于模拟电路。

(2) 控制电路(开关):主要用于数字电路。

晶体管会通过改变输入端(微弱)的电流而影响输出端电压的信号的开关。因此说一般晶体管是电流放大作用,市面上晶体管有两种:

(1) NPN;

(2) PNP。

差别在工作偏压电源用正或负,正电源用NPN,负电源用PNP,而PNP传感器输出级

是电流源型,NPN 传感器输出级是灌电流型。NPN 传感器是电流型需要外接电源,PNP 型电流源就不需要外部电源。

如果发现树莓派板子的电压是 5V 或 3V,但实际接上多个电动机后,常常发生电动机转不动的情况,这是因为树莓派电流过小,那就可以用 NPN 晶体管,使用外接电源,控制另一个电力驱动的直流电动机。如果是直流电动机的电压要很大,或者是交流电的电动机,建议用 Relay 继电器来控制。NPN 晶体管外观与功能如图 19-2 所示。

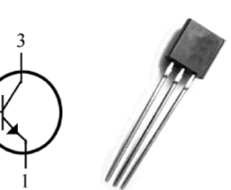




图 19-2　NPN 晶体管作用示意图

通过改变输入端(微弱)的电流而影响输出端电压的信号强弱,因此说(一般)晶体管是电流放大作用。

常见的 NPN 晶体管有

(1) MJE340;

(2) 2N3440;

(3) 2N4401;

(4) 2N3904。

常见的 PNP 晶体管有

(1) 2N4403;

(2) 2N2905A;

(3) 512-PN2907BU。

引脚功能如图 19-2 所示。

(1) 集电极(Collector);

(2) 基极(Base);

(3) 发射极(Emitter)。

不同的 NPN 晶体管的引脚功能有所不同,参考表 19-1。

表 19-1　NPN 晶体管的引脚

NPN 编号	最左边的引脚面对向上	中边的引脚	最右边的引脚
2N3904	发射极 (Emitter)	基极 (Base)	集电极 (Collector)
BC547 和 BC337	集电极 (Collector)	基极 (Base)	发射极 (Emitter)
2N3904 和 2N2222	发射极 (Emitter)	基极 (Base)	集电极 (Collector)
BC 638	发射极 (Emitter)	集电极 (Collector)	基极 (Base)

2. 设备

(1) 树莓派开发板;

(2) 1 个直流电动机(如果没有可以用 LED 代替);

(3) 1 个 NPN 晶体管(作者用的是 2N3904);

(4) 1kΩ 电阻(棕、黑、红、金);

(5) 1 个 1N4001 diode 二极管;

(6) 面包板;

(7) 接线若干。

3. 接线

参考表 19-2 和图 19-3。

表 19-2 硬件接线表

树莓派引脚	说　明
Pin 7 (GPIO4)	晶体管的 1kΩ 电阻,电阻另外接到晶体管的基极
GND	晶体管发射极
无	直流电动机接地接到晶体管集电极和 1N4001 diode 二极管
无	直流电动机电源和 1N4001 diode 二极管

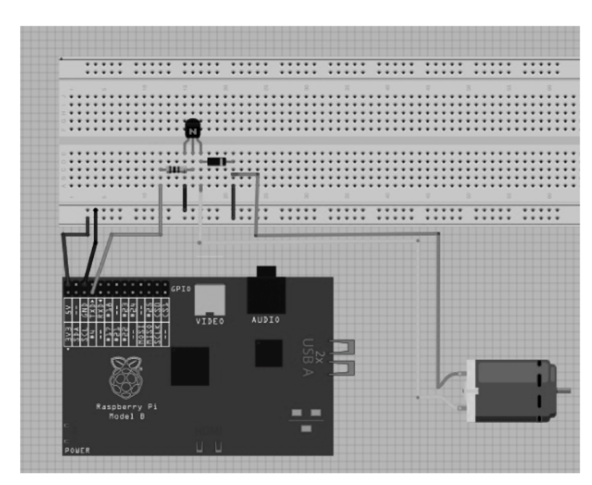


图 19-3 PNP 晶体管的接线图

本案例可以用最基本的 GPIO 数字输出控制,用来测试晶体管的开关操作是否对电动机的控制有效。

样例程序:npn.py

```
1.   #!/usr/bin/env python
2.   # author: Powen Ko
3.   import time,RPi.GPIO as GPIO
4.
5.   GPIO.setmode(GPIO.BOARD)
6.   GPIO.setup(7,GPIO.OUT)
7.
8.   while True:
9.           GPIO.output(7,0)
10.          time.sleep(3)                          #停 3 秒
11.            GPIO.output(7,1)
12.  time.sleep(3)                                  #停 3 秒
```

程序解说

第 6 行：设置引脚 7 为数字输出。

第 8～12 行：无限循环。

第 9 行：把引脚 7 输出低电位。

第 11 行：把引脚 7 输出高电位。

4. 结果

执行这个应用可以看到电动机会运转 3 秒钟，然后停止 3 秒钟，并且一直重复，如图 19-4 所示。

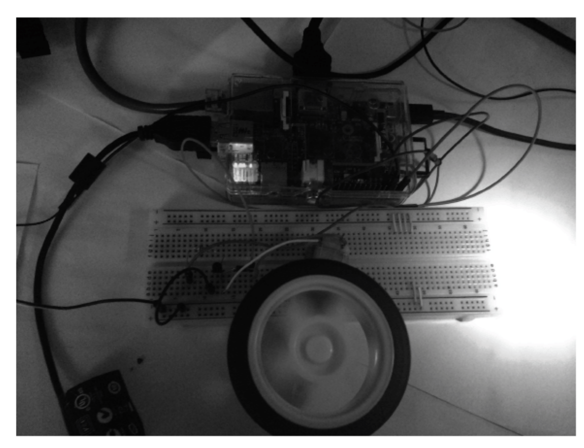


图 19-4　实际执行效果

5. 教学影片

完整的教学视频可以看 RaspberryPi_Car_1_NPN_Morotr-T。

19.2　控制转速

案例 98

1. 介绍

如何通过一个循环来控制输出的电压 PWM，然后连接到 NPN 晶体管和电动机，通过改变电压的方法来调整直流电动机的转速。

2. 准备

(1) 树莓派开发板；

(2) 1 个直流电动机(如果没有可以用 LED 代替)；

(3) 1 个 NPN 晶体管(作者用的是 2N3904)；

(4) 1kΩ 电阻(棕、黑、红、金)；

(5) 1 个 1N4001 diode 二极管；

(6) 面包板;
(7) 接线。

3. 接线

同案例 97。

样例程序:npnspeed.py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
3.  import time,RPi.GPIO as GPIO
4.  GPIO.setmode(GPIO.BCM)
5.  GPIO.setup(4,GPIO.OUT)
6.
7.  p1 = GPIO.PWM(4,1000)                  # channel = 4 frequency = 1000Hz
8.  p1.start(0)
9.
10. while True:
11.     for dc in range (5,101,5):         # 数值改变
12.         p1.ChangeDutyCycle(dc)         # PWM 的输出
13.         time.sleep(0.2);
```

程序解说

第 11 行:读入可变电阻来调整输出的 PWM。

第 12 行:模拟输入数值范围是 0~255,PWM 输出范围是 0~1023,所以通过 map 转换等比。

第 13 行:PWM 的输出。

4. 结果

执行后,可以看到通过程序调整 PWM 的输出,就可以调整直流电动机的转速了,如图 19-5 所示。

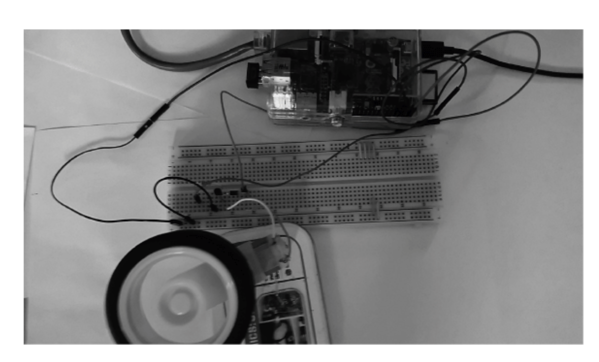


图 19-5 执行结果

5. 教学影片

完整的教学视频可以看 RaspberryPi_Car_2_NPN_MorotrSpeed-T。

19.3 控制电动机 IC L293D——树莓派小汽车前进、后退、旋转

案例 99

1. 介绍

本案例使用到 L293D 电动机驱动 IC,如图 19-6 所示,这是控制电动机很常见的 IC,可以节省电路和同时控制两个电动机进行正向和倒置,也就是可以控制车子的前进、后退与旋转,并且可以控制速度。

建议在实际使用时,L293D 电动机驱动 IC 上面加个散热片,因为 IC 很容易发烫,千万不要用手触摸。

L293D 有 16 引脚,通过内部电路产生信号,电路的输入可以用来设置电动机转动方向,并且可以用于脉宽调整(PWM),也就可以调整电动机速度。另外,L293D 可以同时控制两个直流电动机,如图 19-7 所示,表 19-3 是 L293D 电动机驱动 IC 的引脚菜单。



图 19-6 L293D 电动机驱动 IC 的外观

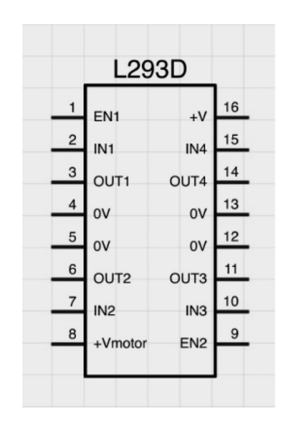


图 19-7 L293D 电动机驱动 IC 的引脚图

表 19-3 L293D 电动机驱动 IC 的引脚菜单

引脚号码	说明	引脚号码	说明
1,EN1	控制第一个电动机的速度	16,+V	接上 5V 直流电压
2,IN1	控制输出 OUT1 的反应	15,IN4	控制输出 OUT4 的反应
3,OUT1	接到第一个电动机的正极	14,OUT4	接到第二个电动机的正极
4,0V	接地	13,0V	接地
5,0V	接地	12,0V	接地
6,OUT2	接到第一个电动机的负极	11,OUT3	接到第二个电动机的负极
7,IN2	控制输出 OUT2 的反应	10,IN3	控制输出 OUT3 的反应
8,+V motor	给电动机的电压,推荐用 12V 直流电压	9,EN2	控制第二个电动机的速度

L293D 电动机驱动操作原理

第 1 个马达需要 3 个控制引脚，IN1、IN2 为电动机转动方向控制信号，当 IN1、IN2 分别为高电位和低电位时电动机正向转，当 IN1、IN2 分别为低电位和高电位时电动机倒置。EN1 引脚使用 PWM 的输入，通过调整 PWM 的占空比(高低电位所占百分比)可以调整电机的转速。

此 IC 可以控制两个马达，引脚 9～15 是控制第 2 个电动机的引脚。

由于不同模块的引脚差异过大，为了方便开发者使用，本案例还是用 L293D IC 直接在面包版上接线，如图 19-8 所示。

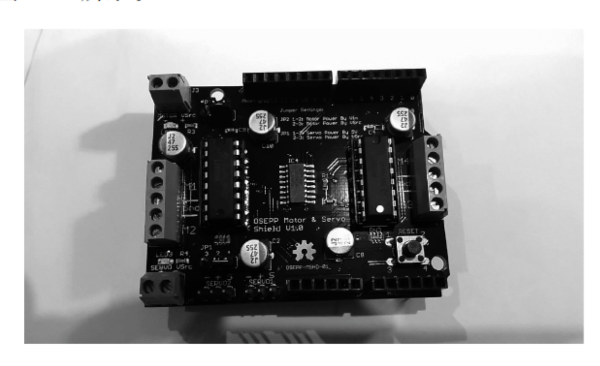


图 19-8　L293D 的板子(OSEPP Motor Shield)

本案例为了让各位轻松的入手，使用 L293D 芯片，请控制电动机的顺时钟和逆时钟的旋转。

2. 设备

(1) 树莓派开发板；

(2) 1 个直流电动机，能有两个更好。

(3) 面包板；

(4) 接线若干；

(5) L293D 电动机驱动 IC；

(6) 5～12V 的电池。

3. 接线

因为 IC 连接到树莓派的线路较为复杂，所以这次接 1 个电动机用来练习控制 L293D 电动机驱动 IC 的操作，如图 19-9 所示，在接线的时候，留意电池正极和负极，如果接反会让 IC 不能提供电力给电动机运转。

样例程序：L293D. py

```
1.  #!/usr/bin/env
2.  #__author__ = "Powen Ko"
3.  import RPi.GPIO as GPIO
```

```
4.  from time import sleep
5.
6.  GPIO.setmode(GPIO.BOARD) # GPIO.BCM
7.
8.  MotorIN1 = 11 # 17
9.  MotorIN2 = 12 # 18
10. MotorEN1 = 13 # 21
11.
12. GPIO.setup(MotorIN1,GPIO.OUT)              #设置3个输出的引脚
13. GPIO.setup(MotorIN2,GPIO.OUT)
14. GPIO.setup(MotorEN1,GPIO.OUT)
15.
16. print "Turning motor on Clockwise."        #电动机顺时针旋转时,需调整这3个引脚电位
17. GPIO.output(MotorIN1,GPIO.HIGH)
18. GPIO.output(MotorIN2,GPIO.LOW)
19. GPIO.output(MotorEN1,GPIO.HIGH)
20.
21. sleep(5)
22. print "Turning motor on Counterclockwise." #马达逆时针旋转时,需调整这3个引脚电位
23. GPIO.output(MotorIN1,GPIO.LOW)
24. GPIO.output(MotorIN2,GPIO.HIGH)
25. GPIO.output(MotorEN1,GPIO.HIGH)
26.
27. sleep(5)
28. print "Stopping motor"
29. GPIO.output(MotorEN1,GPIO.LOW)             #关闭电动机的操作
30. GPIO.cleanup()                             #离开程序前,先恢复GPIO的引脚
31.
32.
```

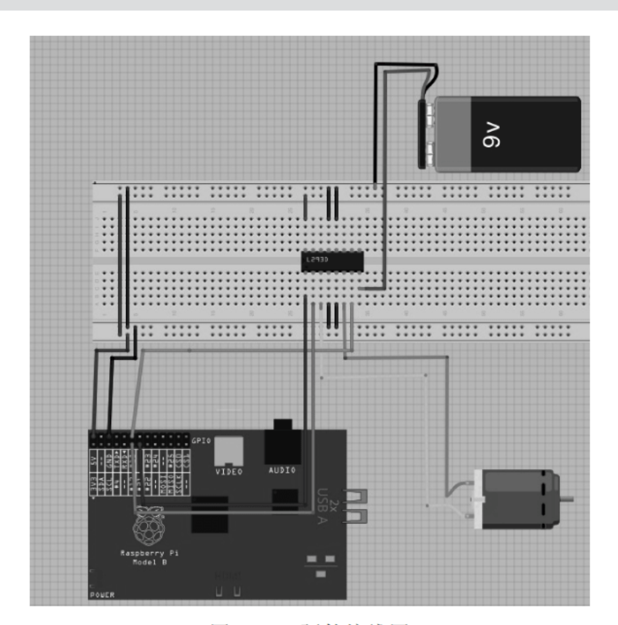


图 19-9　硬件接线图

程序解说

第 8～10 行：设置 3 个输出的引脚，用来控制 IC 的操作。

第 16～19 行：调整这 3 个引脚电位，让电动机顺时针旋转。

第 16～19 行：调整这 3 个引脚电位，让电动机逆时针旋转。

第 29～30 行：离开时的处理操作。

4. 结果

先使用 1 个直流电动机，输入以下命令执行 Python 程序。

```
$ sudo python l293d.py
```

执行的效果会看到电动机先朝一个方向转动 5 秒钟，再转向另外一个方向持续 5 秒钟。如果发现电动机旋转方向相反，把电动机的两个引脚相互对调就可以，如图 19-10 所示。

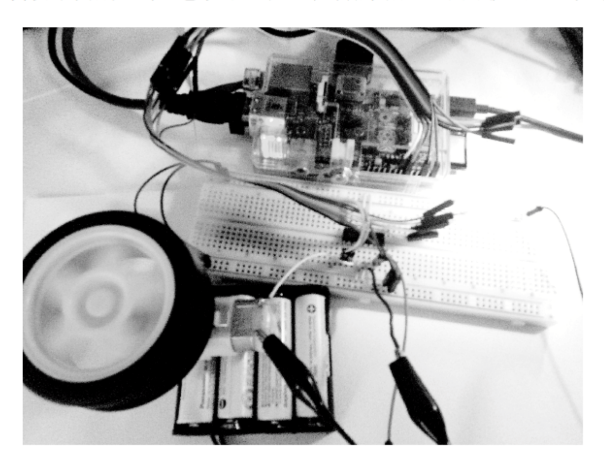


图 19-10 执行效果

5. 教学影片

完整的教学视频可以看 raspberryPi_L293D_1_OneMotor。

19.4 控制电动机转速——树莓派小汽车速度

案例 100

1. 介绍

本案例样同样也是控制电动机驱动 IC L293D，但是和案例 99 不同的是，本案例会加上 PWM 的控制方法调整电动机的转速，也就是说树莓派小汽车跑的速度可以有快慢的调整功能。

出现在 IC 上面的第 1 个引脚 EN1，除了能够控制电动机的开关，如果把它改成使用 PWM 的输入，能够通过调整 PWM 的占空比控制电动机的转速。

2. PWM 函数介绍

脉宽宽度调制 PWM(Pulse Width Modulation)是利用微处理器的数字输出来仿真电路进行控制的一种非常有效的技术,广泛应用于测量、通信到功率控制与变换的许多领域。

PWM 是将信号编码应于脉波宽度上的一种技术,此技术以数字方式来仿真模拟信号,广泛应用在数据传输上。因数字信号只存在 High、Low 电位的变化,相比于模拟信号不会受到噪音干扰。PWM 信号中脉波宽度在整个周期所占的比例称为工作周期(duty cycle),是指位于逻辑高准位(logic high level)的波形在整个周期中占所的比例。PWM 也可以对脉冲的宽度进行调配来有效地获得所需要的波形和电压,脉冲宽度调制是一种模拟时间高低控制方式。

树莓派上的 PWM 功能是通过软件仿真出来的,通过 GPIO. PWM 就能够达到这样的目的。

```
GPIO.PWM(channel,frequency)
```

该函数实际执行的原理是通过自定的频率。

channel:引脚号码;

frequency:频率单位是 Hz。

样例

设置引脚 4 的频率为 100Hz

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
p1 = GPIO.PWM(4,100)
GPIO.start(dc)
```

用于程序引导 PWM 时使用。

dc:duty cycle 范围在 0.0～100.0,意思是高电压的时间在这个频率中占多少。

样例

设置引脚 4 的频率为 100Hz,并且一半是高电压,一半是低电压。

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
p1 = GPIO.PWM(4,100)
p1.start(50)
GPIO.ChangeFrequency(frequency)
```

程序中实时改变 frequency 频率单位是 Hz。

frequency:频率单位是 Hz。

样例

设置引脚 4 的频率为 100Hz,由 100%都是高电压,变换到频率为 20Hz。

```
import RPi.GPIO as GPIO
```

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(12,GPIO.OUT)
p1 = GPIO.PWM(4,50)
p1.start(50)
p1.ChangeFrequency (20)
GPIO.ChangeDutyCycle(dc)
```

dc：duty cycle 范围在 0.0～100.0，意思是高电压的时间在这个频率中占多少。

程序中实时改变 duty cycle 范围在 0.0～100.0。

样例

设置引脚 4 的频率为 100Hz，由 100%都是高电压，变换到一半是高电压，一半是低电压。

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(12,GPIO.OUT)
p1 = GPIO.PWM(4,100)
p1.start(50)
p1.ChangeDutyCycle(50)
```

如果设置 PWM 是 500Hz，图 9-11 中的绿色直线的间隔是 2ms，这是 PWM 的极限。

所以当通过 GPIO.start(dc)函数指定 dc 参数值在 0～100，GPIO.start(100)就是 100%都是高电压，也就造成 3.3V 的效果，然后 GPIO.start(100/2)会造成 1.65V 的效果，如图 19-11 所示。

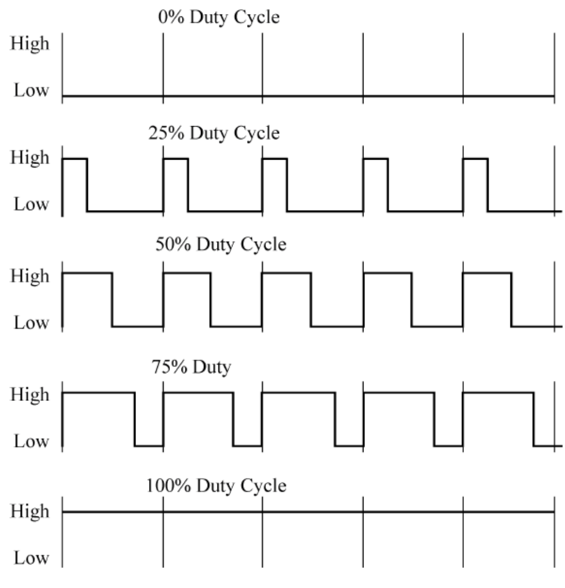


图 19-11　PWM 输出占空比会影响的输出情况

如果有机会用视波器进行输出，就会如图 19-12 所示，是高电压 3.3V 和 0V 之间的快速转换，通过时间差比例，才仿真出模拟输出的效果。



图 19-12　PWM 在视波器上执行的情况

3. 设备

(1) 树莓派开发板；

(2) 1 个直流电动机，如果有两个就更好了；

(3) 面包板；

(4) 接线若干；

(5) L293D 电动机驱动 IC；

(6) 12V～5V 的电池。

4. 接线

同案例 99。

5. 步骤

STEP1：硬件接线

首先参考 19.3 节连接好硬件线路。

STEP2：编写程序

在文本编辑器中把程序内容修改如下，调整的部分是把 EN1 的控制引脚，通过 PWM 方式重新编写，用循环的方法改变 PWM 的占空值来改变电动机转速。

样例程序：L293D_pwm.py

```
1.   #!/usr/bin/env
2.   #__author__ = "Powen Ko"
3.   import RPi.GPIO as GPIO
4.   from time import sleep
```

```
5.
6.  GPIO.setmode(GPIO.BOARD)                        # GPIO.BCM
7.
8.  MotorIN1 = 11                                   # 17
9.  MotorIN2 = 12                                   # 18
10. MotorEN1 = 13                                   # 21
11.
12. GPIO.setup(MotorIN1,GPIO.OUT)                   #设置 3 个输出的引脚
13. GPIO.setup(MotorIN2,GPIO.OUT)
14. GPIO.setup(MotorEN1,GPIO.OUT)
15. p1 = GPIO.PWM(MotorEN1,250)                     # PWM 的频率 = 250Hz
16. p1.start(0)                                     #执行 PWM
17.
18. print "Turning motor on Clockwise."
19. GPIO.output(MotorIN1,GPIO.HIGH)
20. GPIO.output(MotorIN2,GPIO.LOW)
21. for dc in range (0,101,5):
    for(dc = 0;dc < 101;dc++5)
22.   p1.ChangeDutyCycle(dc)                        # 调整 EN1 的 PWM 比例
23.   sleep(1)
24.
25.
26. print "Turning motor on Counterclockwise."      #逆时针处理
27. GPIO.output(MotorIN1,GPIO.LOW)
28. GPIO.output(MotorIN2,GPIO.HIGH)
29. for dc in range (0,101,5):
30.   p1.ChangeDutyCycle(dc)
31.   sleep(1)
32.
33. print "Stopping motor"
34. GPIO.output(MotorEN1,GPIO.LOW)                  #关闭电动机的动作
35. GPIO.cleanup()                                  #离开程序前,先恢复 GPIO 的引脚
```

程序解说

第 8~10 行:设置 3 个输出的引脚,用来控制 IC 的操作。

第 15~16 行:设置 EN1 为 PWM 调整这 3 个引脚电位,让电动机顺时针旋转。

第 19~20 行:调整这两个接脚电位,让电动机顺时针旋转。

第 21~23 行:用循环的方法改变 PWM 的占空值,来调整电动机转速。

第 27~28 行:调整这两个引脚电位,让电动机逆时针旋转。

第 29~31 行:用循环的方法改变 PWM 的占空值,来调整电动机转速。

第 34~35 行:离开时的处理操作。

6. 结果

同样先用 1 个直流电动机,输入以下命令执行 Python 程序。

```
$ sudo python l293d_pwm.py
```

执行的效果如图19-13所示,会看到电动机先朝一个方向,每1秒钟调整转速的快慢,再转向另外一个方向,再做一次转速从慢到快。

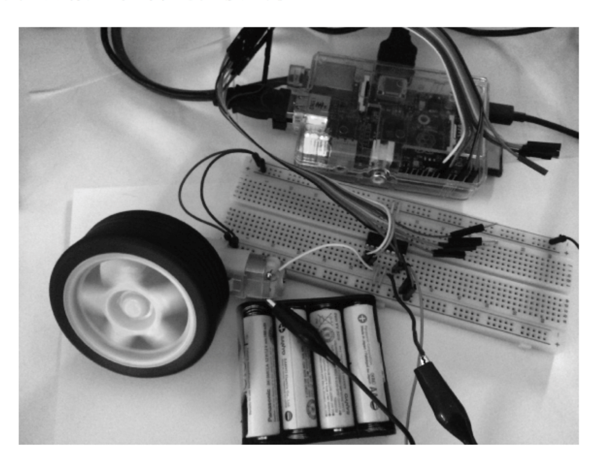


图19-13 执行效果

7. 教学影片

完整的教学视频可以看 raspberryPi_L293D_2_OneMotor_speed.mov。

19.5 控制两个电动机转速——树莓派小汽车

案例101

1. 介绍

本案例通过Python和L293D的控制IC来引导两个电动机,并且分享从无到有的自制树莓派小汽车的例子。

小汽车的部分要怎样处理就看各位的预算,可以在市面上买很多小汽车的零件,如果要追求DIY,可以把家里没在玩的小汽车重新组装,如果想用最低的预算达到目的,可以用木板当车子的底座,加两个电动机和轮子。如果只是练习此案例,没有必要去购买这些小汽车的零件,执行程序练习一下就可以了。注意直流电动机的额定功率不要超过12V,因为L293D最多支能到12V电动机。

2. 设备

(1) 树莓派开发板;

(2) 1个直流电动机,如果有两个就更好了。

(3) 面包板;

(4) 接线若干;

(5) L293D 电动机驱动 IC;

(6) 2 个 12～5V 的电池;

(7) 小汽车零组件。

3. 接线

这里的电子硬件和案例 100 很像,不一样的地方是要控制两个电动机,树莓派的电源请接到智能手机的便携充电器上,如图 19-14 所示。

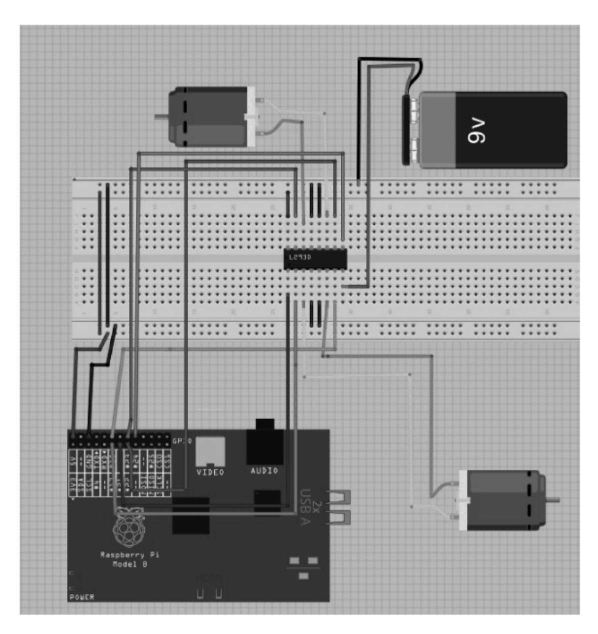


图 19-14 硬件接线

4. 汽车模块

现在各大拍卖网站上,只要搜索 arduino 零件都容易买到,这样的硬件也适用于树莓派上。本案例的小汽车就是这样获取的。如果有 3D 打印机,也可以自己下载和制造,像 Arduino Controlled Servo Robort 就可以达到这样的需求,能免费下载 3D A 模型文档 STL,如图 19-15 所示。网址为:http://www.thingiverse.com/thing:209。

5. 步骤

STEP1:小汽车组装

首先把小汽车零件进行组装,如图 19-16 所示的图片是作者在网络上购买的电子小汽车零件自行组装而成,过程因为每一个组件差异太大,所以就不多说了,请按照实际情况自行处理。如果没有,也可以用两个电动机来测试。

STEP2:预先测试

以作者的经验在处理较复杂的应用时,先在实际的树莓派上测试硬件和过程,才会跟其



图 19-15　3D 打印的小汽车模块

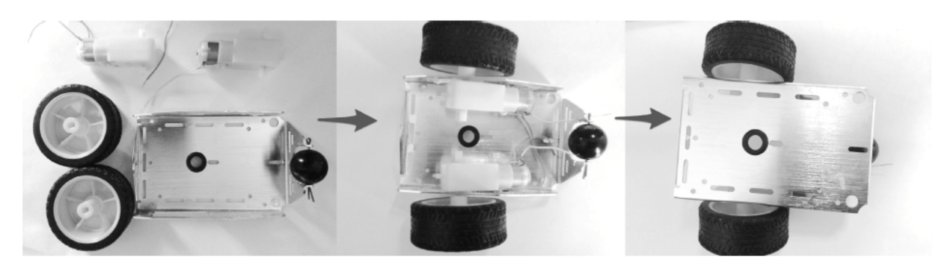


图 19-16　小汽车组装

他的硬件作组装，所以请先用面包板把线路和硬件组装好，不用急着放上小汽车，跟软件测试成功之后再放上小汽车。提醒组装硬件线路时，先把树莓派的电源关闭。

STEP3：硬件接线

在文本编辑器中把程序内容修改如下，调整的部分是把另外一边的电动机控制线路一起处理好。

样例程序：l293d_2_pwm.py

```
1.  #!/usr/bin/env
2.  #__author__ = "Powen Ko"
3.  import RPi.GPIO as GPIO
```

```
4.  from time import sleep
5.
6.  GPIO.setmode(GPIO.BOARD)                          # GPIO.BCM
7.
8.  MotorIN1 = 11                                     # 17
9.  MotorIN2 = 12                                     # 18
10.  MotorEN1 = 13                                    # 21
11.
12.  MotorIN3 = 15                                    # 22
13.  MotorIN4 = 16                                    # 23
14.  MotorEN2 = 18                                    # 24
15.
16.  GPIO.setup(MotorIN1,GPIO.OUT)                    # 设置 6 个输出的引脚
17.  GPIO.setup(MotorIN2,GPIO.OUT)
18.  GPIO.setup(MotorEN1,GPIO.OUT)
19.  GPIO.setup(MotorIN3,GPIO.OUT)
20.  GPIO.setup(MotorIN4,GPIO.OUT)
21.  GPIO.setup(MotorEN2,GPIO.OUT)
22.
23.  p1 = GPIO.PWM(MotorEN1,250)                      #PWM 的频率为 250Hz
24.  p1.start(0)                                      # 执行 PWM
25.  p2 = GPIO.PWM(MotorEN2,250)                      # PWM 的频率为 250Hz
26.  p2.start(0)                                      # 执行 PWM
27.
28.  print "Turning motor on Clockwise."
29.  GPIO.output(MotorIN1,GPIO.HIGH)
30.  GPIO.output(MotorIN2,GPIO.LOW)
31.  GPIO.output(MotorIN3,GPIO.HIGH)
32.  GPIO.output(MotorIN4,GPIO.LOW)
33.  for dc in range (0,101,5):
     for(dc = 0;dc < 101;dc ++ 5)
34.     p1.ChangeDutyCycle(dc)                        #调整 EN1 的 PWM 比例
35.     p2.ChangeDutyCycle(dc)
36.     sleep(0.2)
37.
38.  print "Turning motor on Counterclockwise."       #逆时针处理
39.  GPIO.output(MotorIN1,GPIO.LOW)
40.  GPIO.output(MotorIN2,GPIO.HIGH)
41.  GPIO.output(MotorIN3,GPIO.LOW)
42.  GPIO.output(MotorIN4,GPIO.HIGH)
43.  for dc2 in range (0,101,5):
44.     p1.ChangeDutyCycle(dc2)
45.     p2.ChangeDutyCycle(dc2)
46.     sleep(0.2)
47.
48.  print "Stopping motor"                           # 关闭电动机的操作
49.  GPIO.output(MotorEN1,GPIO.LOW)
50.  GPIO.output(MotorEN2,GPIO.LOW)
51.  GPIO.cleanup()                                   # 离开程序前,先恢复 GPIO 的引脚
```

程序解说

第 16～21 行：设置 6 个输出的引脚，用来控制 IC 的操作。

第 23～26 行：设置 EN1 和 EN2 为 PWM 调整电动机转速。

第 29～32 行：调整这 4 个引脚电位，让两个电动机顺时针旋转。

第 33～36 行：以循环的方法改变 PWM 的占空值，来调整电动机转速。

第 39～42 行：调整这 4 个引脚电位，让两个电动机逆时针旋转。

第 43～46 行：以循环的方法改变 PWM 的占空值，来调整电动机转速。

第 49～51 行：离开时的处理操作。

6. 结果

本案例用两个直流电动机，输入以下命令执行 Python 程序。

```
$ sudo python l293d_2_pwm.py
```

执行效果如图 19-17 所示，两个电动机一起朝一个方向由慢到快的旋转，再一起转向另外一个方向，再做一次从慢到最快的旋转。

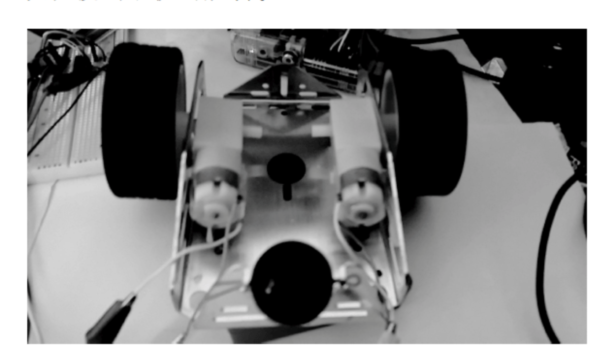


图 19-17　执行效果

如果电动机没有转，留意一下是否电动机功率不足，导致转不动。如果发生这种情况，更换大功率的电池就会有帮助。

7. 教学影片

完整的教学视频可以看 raspberryPi_L293D_3_2Motors。

19.6　无线遥控小汽车

案例 102

1. 介绍

本案例会通过键盘来控制树莓派的 GPIO，这样就可以实时控制 IC L293D 来引导两个电动机。过程比较复杂，有几个问题需要先解决，无线遥控和把树莓派放到小车子上，会发

现想把树莓派放在车子上面,还有几条线需要拿掉,例如

(1) HDMI 线;

(2) 网线;

(3) 树莓派的电源线。

本案例会为这些部分找到解决方法。

2. 设备

(1) 树莓派开发板;

(2) 2 个直流电动机;

(3) 面包板;

(4) 接线若干;

(5) L293D 电动机驱动 IC;

(6) 2 个 12V~5V 的电池;

(7) 小汽车零组件;

(8) WiFi USB Dongle;

(9) USB 无线键盘;

(10) 5V 直流电和 2A 以上的智能手机的便携充电器。

3. 接线

同案例 101。

4. 步骤

首先完成“19.5 控制两个电动机转速——树莓派小汽车”。

STEP1:编写程序

在文本编辑器中把程序内容修改如下,主要调整部分是通过键盘的按键改变电动机的运转方法,本案例特意为树莓派小汽车设计了以下操作:

(1) s 键:电动机暂停运转。

(2) q 键:离开程序。

(3) 1 键:调整电动机转速最快。

(4) 2 键:调整电动机转速适中。

(5) 3 键:调整电动机转速最慢。

(6) w 键:小汽车往前跑。

(7) x 键:小汽车往后跑。

(8) a 键:右转。

(9) d 键:左转。

样例程序:l293d_remote.py

```
1.    #!/usr/bin/env
2.    #__author__ = "Powen Ko"
```

```
3.  import RPi.GPIO as GPIO
4.  from time import sleep
5.  import threading
6.  import sys
7.  import sys,tty,termios
8.
9.  def getch(): # 键盘处理
10.     fd = sys.stdin.fileno()
11.         old = termios.tcgetattr(fd)
12.         try:
13.             tty.setraw(fd)
14.             return sys.stdin.read(1)
15.     finally:
16.         termios.tcsetattr(fd,termios.TCSADRAIN,old)
17.
18. class KeyEventThread(threading.Thread):                # 多执行线程
19.     def run(self):
20.         print("thread");
21.         Fun()
22.
23. def Fun():
24.     print("Fun")
25.     while True:
26.         key = getch()                                   # 按键处理
27.         if key == 'q':
28.             funExit()
29.             exit()
30.             return
31.         elif key == '1':
32.             print('speed 1')
33.             funSpeed(100,100)
34.         elif key == '2':
35.             print('speed 2')
36.             funSpeed(70,70)
37.         elif key == '3':
38.             print('speed 3')
39.             funSpeed(20,20)
40.         elif key == 'w':
41.             print('forward')                            #往前跑
42.             GPIO.output(MotorIN1,GPIO.HIGH)
43.             GPIO.output(MotorIN2,GPIO.LOW)
44.             GPIO.output(MotorIN3,GPIO.HIGH)
45.             GPIO.output(MotorIN4,GPIO.LOW)
46.             funSpeed(50,50)
```

```
47.             elif key == 'x':                          # 往后跑
48.                 print('backward')
49.                 GPIO.output(MotorIN1,GPIO.LOW)
50.                 GPIO.output(MotorIN2,GPIO.HIGH)
51.                 GPIO.output(MotorIN3,GPIO.LOW)
52.                 GPIO.output(MotorIN4,GPIO.HIGH)
53.                 funSpeed(50,50)
54.             elif key == 'a':                          # 往左跑
55.                 print('left')
56.                 GPIO.output(MotorIN1,GPIO.HIGH)
57.                 GPIO.output(MotorIN2,GPIO.LOW)
58.                 GPIO.output(MotorIN3,GPIO.LOW)
59.                 GPIO.output(MotorIN4,GPIO.HIGH)
60.                 funSpeed(50,50)
61.             elif key == 'd':                          # 往右跑
62.                 print('right')
63.                 GPIO.output(MotorIN1,GPIO.LOW)
64.                 GPIO.output(MotorIN2,GPIO.HIGH)
65.                 GPIO.output(MotorIN3,GPIO.HIGH)
66.                 GPIO.output(MotorIN4,GPIO.LOW)
67.                 funSpeed(50,50)
68.             elif key == 's':
69.                 print('stop')
70.                 funSpeed(0,0)
71.             else:
72.                 print("key = " + key)
73.         return
74.
75.
76.     def funSpeed(i1,i2):
77.         dc1 = i1
78.         dc2 = i2
79.         p1.ChangeDutyCycle(dc1)
80.         p2.ChangeDutyCycle(dc2)
81.
82.
83.     def funInit():
84.         GPIO.setmode(GPIO.BOARD)                      # 设置 6 个输出的引脚
85.         GPIO.setup(MotorIN1,GPIO.OUT)
86.         GPIO.setup(MotorIN2,GPIO.OUT)
87.         GPIO.setup(MotorEN1,GPIO.OUT)
88.         GPIO.setup(MotorIN3,GPIO.OUT)
89.         GPIO.setup(MotorIN4,GPIO.OUT)
90.         GPIO.setup(MotorEN2,GPIO.OUT)
91.
92.     def funExit():                                    # 关闭电动机的动作
```

```
93.         print "Stopping motor"
94.         GPIO.output(MotorEN1,GPIO.LOW)
95.         GPIO.output(MotorEN2,GPIO.LOW)
96.         GPIO.cleanup()
97.
98.
99.      MotorIN1 = 11                          # 17
100.     MotorIN2 = 12                          # 18
101.     MotorEN1 = 13                          # 21
102.     MotorIN3 = 15                          # 22
103.     MotorIN4 = 16                          # 23
104.     MotorEN2 = 18                          # 24
105.
106.     print("Press 'q' to exit")
107.     print("'w' = forward, 'x' = backward, 'a' = left, 'd' = right, 's' = stop")
108.     print("'1', '2', '3' motor speed")
109.     funInit()
110.     p1 = GPIO.PWM(MotorEN1,250)            # PWM 的频率为 250Hz
111.     p1.start(0)
112.     p2 = GPIO.PWM(MotorEN2,250)
113.     p2.start(0)
114.
115.     kethread = KeyEventThread()
116.     kethread.start()
```

程序解说

第 83～90 行：设置 6 个输出的引脚，用来控制 IC 的操作。

第 110～113 行：设置 EN1 和 EN2 为 PWM 调整电动机转速。

第 40～46 行：调整这 4 个引脚电位，让两个电动机顺时针旋转，这样小汽车就会往前跑。

第 47～53 行：调整这 4 个引脚电位，让两个电动机逆时针旋转，这样小汽车就会往后跑。

第 54～60 行：调整这 4 个引脚电位，让两个电动机一正一反，这样小汽车就会往左跑。

第 61～67 行：调整这 4 个引脚电位，让两个电动机一反一正，这样小汽车就会往右跑。

第 9～16 行：读入键盘的数据。

STEP2：测试写

执行 Python 程序，先做测试确认可以通过键盘的按键控制电动机，再把设备搬到小汽车上面。

```
$ sudo python l293d_remote.py
```

STEP3：去掉电源线

树莓派的供电推荐使用智能手机的便携充电器，基本足够让树莓派使用了。使用输出有 5V 直流电和 2A 以上的移动手机电池，如图 19-18 所示。



图 19-18　智能手机的便携充电器

STEP4：去掉网线

树莓派上面也许有网线，按照本书设置 USB WiFi 连接的方法，使用 WiFi USB Dongle 做网络连接。

STEP5：控制

因为要用键盘来控制树莓派小汽车，但希望小汽车可以自由行动，不会因为键盘线卡住，有两个方法可以解决此问题。

（1）使用网络和 SSH 的方法来控制树莓派，这样就不需要在树莓派上连接键盘了。

（2）使用无线键盘进行控制。

作者建议使用 USB 无线键盘，如图 19-19 所示，挑选计算机使用的 USB 无线键盘就可以正常使用，不必另外购买特殊的键盘。



图 19-19　USB 无线键盘

STEP6：屏幕

本案例不需要用 HDMI 线接到屏幕上，可以通过 SSH 网络连接的方法来控制树莓派，所以用计算机或手机的 SSH 都可以。

5. 执行方法和结果

首先关闭树莓派。

(1) 把树莓派放上小汽车。

(2) 把相关电路也放上小汽车，并用绝缘胶带绑紧。

(3) 把 2 个电池(树莓派用的 USB 充电器)放上小汽车。

(4) 把电池分别接上树莓派和 L293D。

(5) 等待树莓派开机。

(6) 可以用计算机或手机的 SSH 连接到树莓派上。

(7) 执行程序。

使用 SSH 和键盘来控制两个直流电动机的前后左右的动作，按照以下的步骤设置：

(1) 在手机或计算机上，通过 SSH 客户端软件连接到树莓派小汽车上。

(2) 在手机上的 SSH 客户端，输入以下命令执行树莓派小汽车上 Python 程序。

```
$ sudo python l293d_remote.py
```

请分别用键盘上的 w、x、d、a、s 键来控制小汽车的前后左右和停止，1、2、3 键来控制速度。如果要离开程序按 q 键。此时就能顺利地控制树莓派小汽车了，如图 19-20 所示。

注意　(1) USB 无线键盘接在 SSH 的计算机端，而不是接在树莓派。

(2) 根据实际测试，部分牌子的移动手机电池不能顺利引导树莓派，请使用输出有 5V 直流电和 2A 以上的移动手机电池。

(3) 多用绝缘胶带把线路和设备固定好。

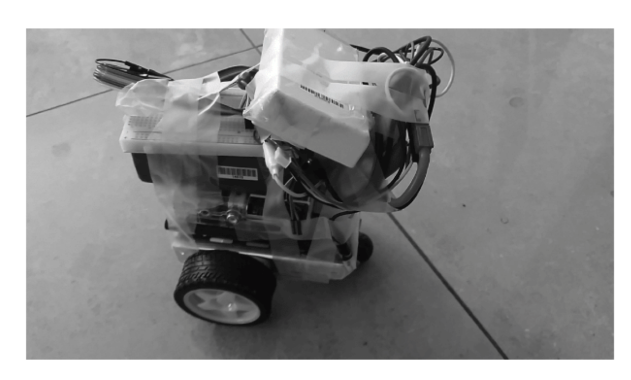


图 19-20　执行效果

6. 执行影片

成品展示的视频可以看 raspberryPi_L293D_4_remote_final.mp4。

7. 教学影片

完整的教学视频可以看 raspberryPi_L293D_4_remote_T。

19.7 控制小汽车

19.7.1 使用 Android 控制树莓派小汽车

案例 103

1. 介绍

现在智能手机到处都有并且人手一机,所以本案例通过智能手机来控制案例 102 自制的树莓派小汽车。

2. 设备

(1) 树莓派开发板;

(2) 2 个直流电动机;

(3) 面包板;

(4) 接线若干;

(5) L293D 电动机驱动 IC;

(6) 2 个 12V~5V 的电池;

(7) 小汽车零组件;

(8) WiFi USB Dongle;

(9) USB 无线键盘;

(10) 5V 直流电和 2A 以上的智能手机的便携充电器;

(11) Android 或者是 iOS 平台的智能手机或平板电脑。

3. 接线

同案例 102。

4. 步骤

首先完成案例 102,因为不同的操作系统需要的设置不太一样,请按照实际情况,挑选以下的章节做出设置,特别提醒 iOS 和 Android 都必须在同一个网络才可以顺利执行,也就是说使用智能手机或者平板电脑,必须要连接到的树莓派所使用的相同 WiFi 路由器上,这样彼此才能顺利沟通,并且把智能手机的 2G、3G、4G 暂时关闭。

首先完成"19.6 无线遥控小汽车",把树莓派小汽车先引导并连接到网络上。

STEP1:WiFi 网络连接

首先把 Android 手机连接到和树莓派相同的 WiFi 无线网络上,如图 19-21 所示。

STEP2：安装 Android 软件——JuiceSSH

JuiceSSH 这款免费的 Android SSH APP，如图 19-22 所示，使用起来相当便利好用。在 Android 操作系统的平板电脑或者是智能手机的 Store 上搜索 JuiceSSH 就可以找到。

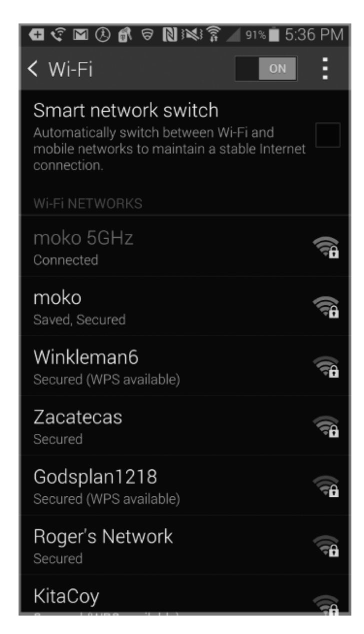


图 19-21　连接到和树莓派相同的 WiFi 无线网络上

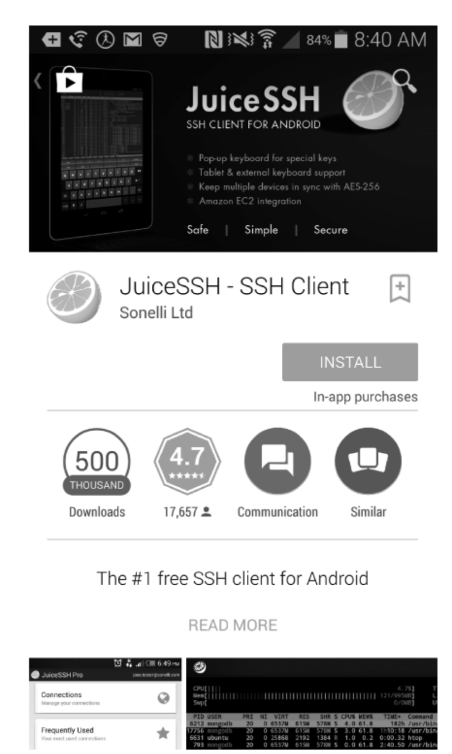


图 19-22　通过 Google Store 下载 JuiceSSH

也可以在网页上，如图 19-23 所示，通过浏览器下载 https://play.google.com/store/apps/details?id=com.sonelli.juicessh&hl=en。

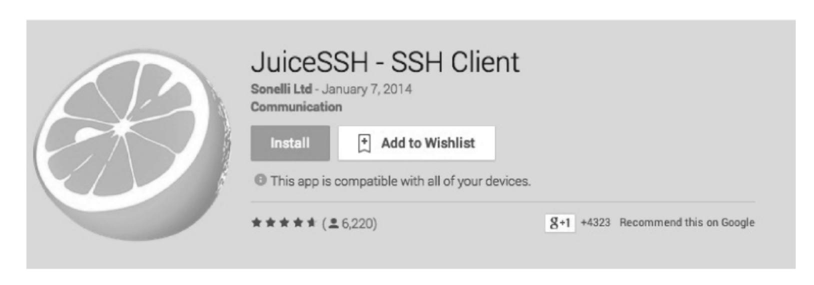


图 19-23　好用的 Android 软件 JuiceSSH-SSH Client

STEP3：安装 Android 软件——JuiceSSH

在 Android 上执行 JuiceSSH，第一次打开时需要作一些设置，新增连接单击 New Connection 按钮，如图 19-24 所示。

STEP4：网络设置的相关数据

输入网络设置的相关数据，如图 19-25 所示。

（1）昵称；

（2）连接方式；

（3）树莓派的网络地址；

（4）账号和密码。

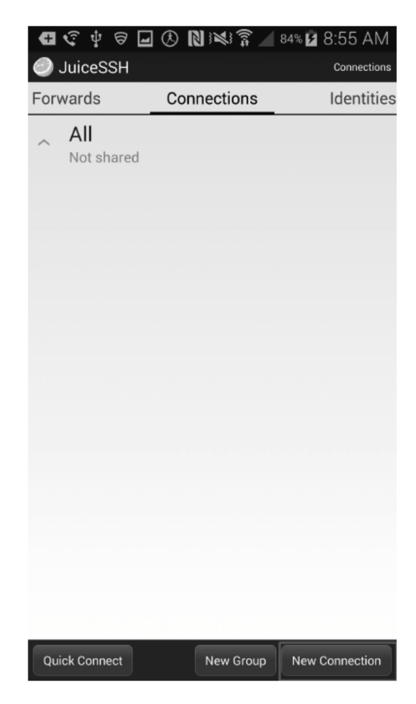


图 19-24　在 Android 上执行 JuiceSSH

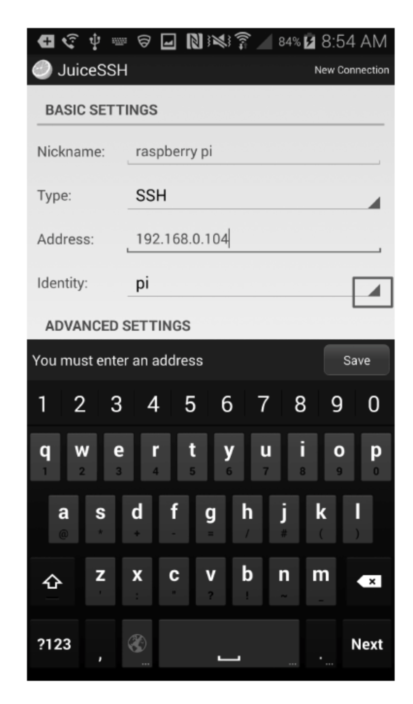


图 19-25　网络设置的相关数据

STEP5：账号密码的设置页面

在 Identity 的账号密码设置页面，把树莓派的 SSH 登录时的账号和密码登记在此，如图 19-26 所示。

（1）Nickname：昵称。

（2）Username：SSH 登录树莓派时的账号，如果之前没有修改树莓派的账号和密码，这里用户名称是 pi。

（3）Password：SSH 登录树莓派时的密码，如果之前没有修改树莓派的账号和密码，这

里用户名称是 raspberry。

(4) Private Key：私有密钥，一般不用修改。

STEP6：连接

把树莓派小汽车的电源打开准备连接，并且确认树莓派遥控小汽车已经顺利连上 WiFi 无线网络，回到 Android 的 JuiceSSH，选中刚刚的设置选项，如图 19-27 所示。如果一切设置没有错误，就可顺利连接到 SSH 文字模式中，如果发现不能连接，请确认配置文件的网络位置是否正确。

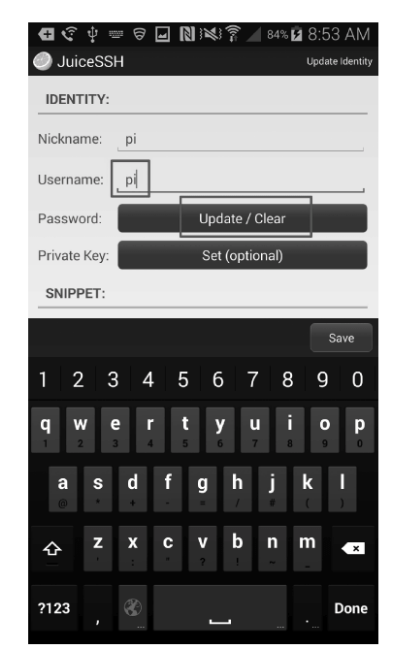


图 19-26　账号密码设置页面



图 19-27　网络设置的相关数据

STEP7：连接到 SSH

通过 Android 手机连接到树莓派，执行同一个样例的 Python 程序，命令如下：

```
$ sudo python l293d_remote.py
```

就可以顺利执行，操作方法也是一样，通过 w、x、a、d、s 按键来控制遥控小汽车，如果要离开按 q 键。执行起来画面如图 19-28 所示。

5. 结果

详细的执行结果可以看视频 raspberryPi_L293D_4_remote_Android，就会看到 Android 智能手机通过 JuiceSSH 远程连接到树莓派去执行 Python 程序，控制树莓派遥控小汽车，如图 19-29 所示。

```
85% 8:57 AM
Linux raspberrypi 3.12.22+ #691 PREEMPT Wed Jun 18 18:29:58 BST 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 16 15:52:21 2014 from 192.168.0.106
pi@raspberrypi ~ $ cd code
pi@raspberrypi ~/code $ sudo python l293d_remote.py
Press 'q' to exit
'w'=forward,'x'=backward,'a'=left,'d'=right,'s'=stop
'1','2','3' motor speed
thread
Fun
```

图 19-28　通过 JuicesSSH 执行 l293d_remote.py



图 19-29　实际执行的情况

19.7.2　使用 iOS 控制树莓派小汽车

案例 104

1. 介绍

案例 103 介绍的是使用 Android 系统的智能手机控制树莓派遥控小汽车，本案例介绍使用苹果的 iOS 操作系统来控制树莓派小汽车。其实这两种操作系统控制树莓派的原理是一样的，都是通过 SSH 软件达到目的。当然有人会好奇，是不是还有其他技术可以让树莓派和智能手机相互沟通？如果各位熟悉 iOS 和 Android 系统的程序开发，就会有很多的方法，例如通过 TCP/IP、网页、蓝牙等技术都可以做到。

2. 步骤

首先完成“19.7.1 使用 Android 控制树莓派小汽车”，把树莓派小汽车先引导并连接到

网络上。

STEP1：WiFi 网络连接

确认 iOS 手机已经连接到和树莓派相同的 WiFi 无线网络上，如图 19-30 所示。



图 19-30　连接到和树莓派相同的 WiFi 无线网络上

STEP2：安装 Android 软件——JuiceSSH

如果想使用 iOS 智能手机从远程 SSH 访问控制，可以使用 WebSSH 这款免费的 APP，使用起来相当便利好用，如图 19-31 所示。可以躺在沙发上，通过手机连接到树莓派做事情，听起来真是不可思议，但现在通过手机版的 SSH 软件，就可以做到。可以在 APP Store 上搜索 SSH 就可以找到。

STEP3：把树莓派遥控小汽车电源打开

和刚刚 Android 的操作步骤一样，把树莓派小汽车的电源打开，如图 19-32 所示，连接和确认树莓派遥控小汽车已经顺利连上同一个局域网的 WiFi 无线网络，并且确认配置文件的网络位置正确。

STEP4：引导 WebSSH

安装结束后，在 iOS 上执行 WebSSH，在第一次打开时，需要做一些设置并且新增连接，单击＋按钮进入连接设置界面，如图 19-33 所示。

STEP5：网络设置的相关数据

接下来填写网络设置的相关数据，如图 19-34 所示。

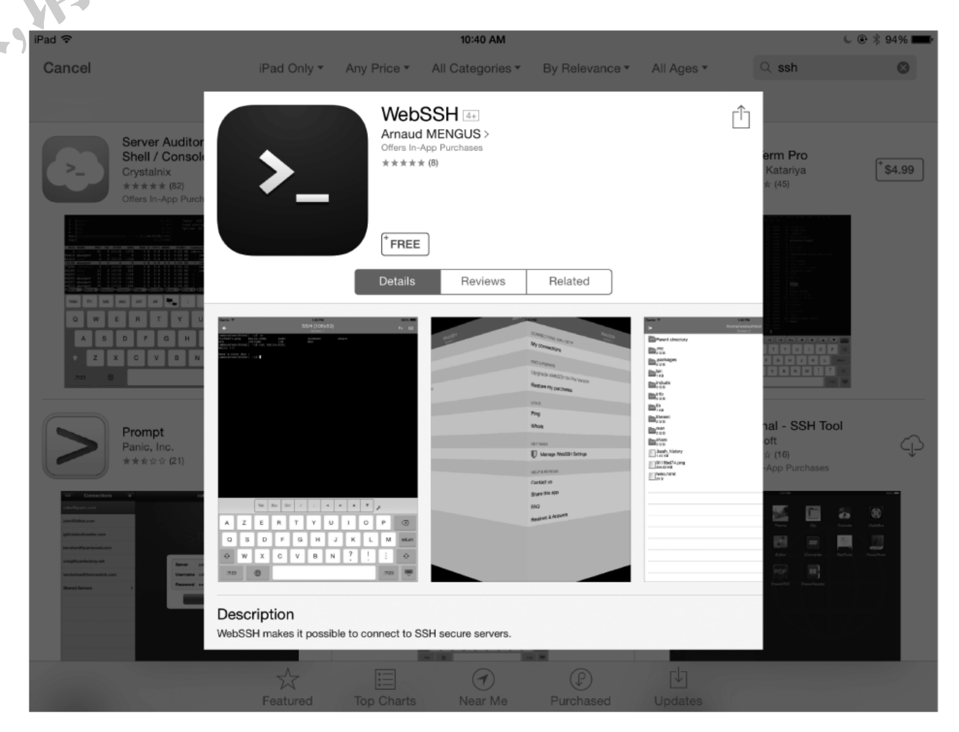


图 19-31　通过 APP Store 下载 WebSSH



图 19-32　把树莓派遥控小汽车电源打开

(1) Host：树莓派的网络地址，按照实际的树莓派网络地址填写。

(2) Port：SSH 连接的端口号，不要修改默认的 22。

(3) User：SSH 登录树莓派时的账号，如果之前没有修改树莓派的账号和密码，这里用户名称是 pi。

(4) Password：SSH 登录树莓派时的密码，如果之前没有修改树莓派的账号和密码，这里用户名称是 raspberry。



图 19-33　在 iOS 上执行 JuiceSSH

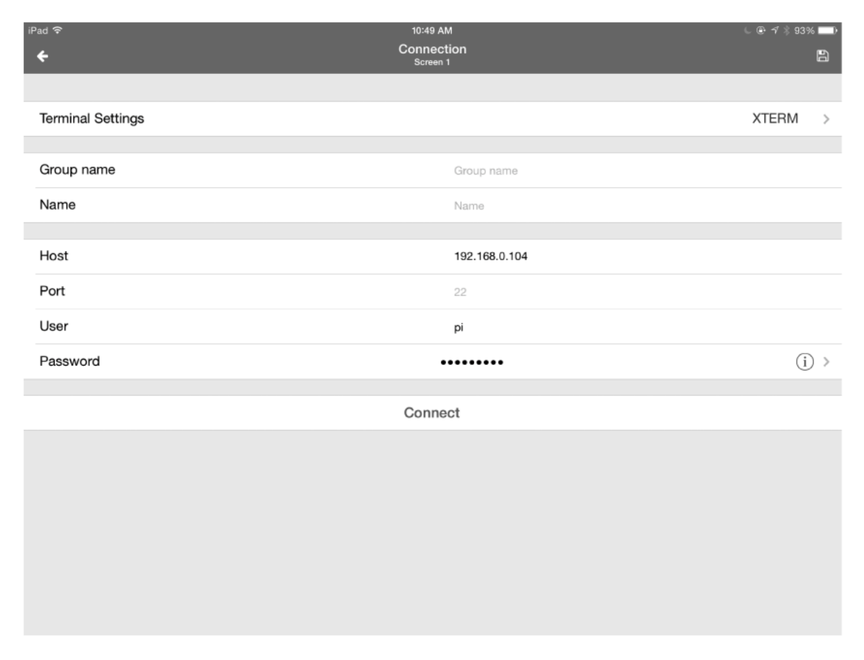


图 19-34　网络设置的相关数据

完成后，单击 Connection 按钮，就可以顺利地通过 SSH 连接到树莓派上。

STEP6：连接到 SSH

当顺利通过 iOS 手机连接到树莓派，也是执行同一个样例的 Python 程序，命令如下：

```
$ sudo python l293d_remote.py
```

控制树莓派遥控小汽车的方法也是一样的，通过 w、x、a、d、s 按键来控制遥控小汽车，如果要离开按 q 键。WebSSH 执行起来画面如图 19-35 所示。



图 19-35　通过 WebSSH 执行 l293d_remote.py

3. 结果

详细的执行结果可以看视频 raspberryPi_L293D_5_remote_iOS，就会看到 iPad 通过 WebSSH 远程连接到树莓派去执行 Python 程序，控制树莓派遥控小汽车，如图 19-36 所示。

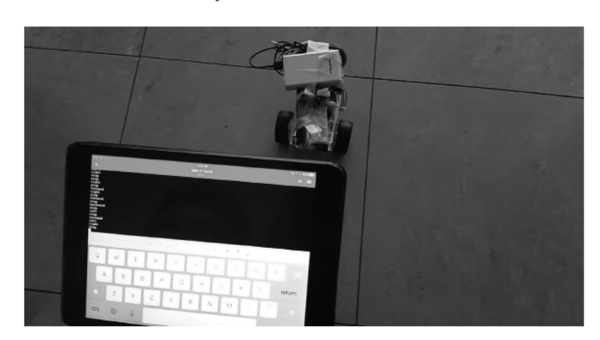


图 19-36　实际执行的情况

第 20 章 树莓派迷你无线路由器

20.1 树莓派无线路由器

案例 105

1. 介绍

本案例介绍如何把树莓派变成一个迷你无线路由器,也称为口袋型无线基地台或 WiFi 网络分享器,也就是如何把 RJ45 的网络,通过 WiFi 的方式分享给平板电脑和手机使用,如图 20-1 是 D-Link 的 Wireless G108 Access Point 无线路由器。各位如果在外住宿,只有一条网络线,想让多个设备同时上网,把树莓派变成一个迷你无线路由器就是一个相当好用的小工具。不想买无线路由器,可以通过网络线分享给带有 WiFi 通讯的设备,通过这项产品让计算机、手机与平板电脑同时上网。



图 20-1 D-Link 的 Wireless G108 Access Point 无线路由器

树莓派迷你无线路由器是一款超迷你、操作极其简单的无线路由器,用户只需把树莓派迷你无线路由器插到调制解调器,不需额外的设置,就能把连接有线网络的树莓派变成接入点,并将 WiFi 信号供手机、iPad 等多台智能设备连接上网!

这个应用难度高,也相当麻烦,原因是 USB WiFi Dongle 的驱动程序问题,本案例使用

的是 RTL8188C 芯片的 WiFi，并且下载驱动程序的源程序来修改这个问题，如果不确定你使用的设备，可以输入 lsusb 查看无线网卡是什么型号。推荐本书使用的硬件 RealTek RTL8188CUS 802.11n，这样此案例会容易一些。

2. 目的

把网络信号分享到 WiFi，并指派 IP 给其他的连接设备。

3. 设备

需要有以下硬件：

(1) 树莓派开发板；

(2) 1 根网线；

(3) 1 个 USB WiFi Dongle。

需要有以下软件：

(1) DHCP；

(2) Python GPIO 函数库。

4. 接线

直接把网线接在树莓派上，把 USB WiFi Dongle 插到 USB 插槽上，并且确认网线的信号可以正常工作。

5. 步骤

STEP1：更新安装软件

输入以下命令，更新安装软件。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

STEP2：安装 DHCP

输入以下命令，安装 hostapd 和 isc-dhcp-server 软件，如图 20-2 所示。

```
$  sudo apt - get install hostapd isc - dhcp - server
```

```
pi@raspberrypi ~ $ sudo apt-get install hostapd isc-dhcp-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  isc-dhcp-server-ldap
The following NEW packages will be installed:
```

图 20-2　安装 hostapd 软件

STEP3：设置/etc/dhcp/dhcpd.conf 的配置文件

用文本编辑器修改/etc/dhcp/dhcpd.conf 的配置文件，如图 20-3 所示。

```
$  sudo nano /etc/dhcp/dhcpd.conf
```

分别把 domain-name 的配置文件删掉,由原本的

```
option domain - name "example.org";
option domain - name - servers ns1.example.org,ns2.example.org;
```

修改为

```
#option domain - name "example.org";
#option domain - name - servers ns1.example.org,ns2.example.org;
```

并把这一个自动设置选项删除掉#,意思是执行此命令。

```
# authoritative;
```

修改为

```
authoritative;
```

```
GNU nano 2.2.6          File: /etc/dhcp/dhcpd.conf          Modified

#
# Sample configuration file for ISC dhcpd for Debian
#
#

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
# option domain-name "example.org";
# option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

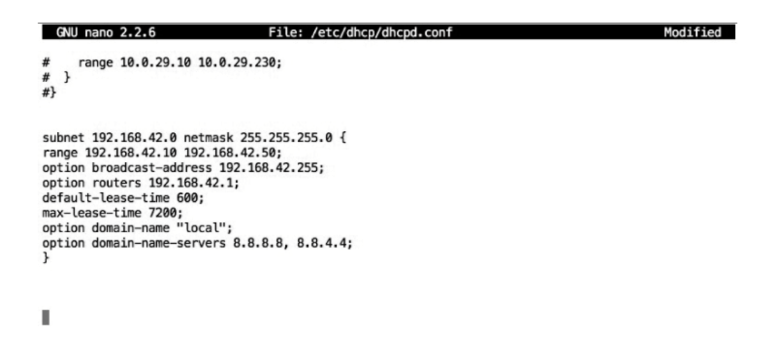
# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

^G Get Help   ^O WriteOut   ^R Read File   ^Y Prev Page   ^K Cut Text     ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is    ^V Next Page   ^U UnCut Text   ^T To Spell
```

图 20-3　修改/etc/dhcp/dhcpd.conf 设置

在这个文件的最后面添加以下设置数据,完毕后进行存储离开文本编辑环境,如图 20-4 所示。

```
subnet 192.168.42.0 netmask 255.255.255.0 {
range 192.168.42.10 192.168.42.50;
option broadcast - address 192.168.42.255;
option routers 192.168.42.1;
default - lease - time 600;
max - lease - time 7200;
option domain - name "local";
option domain - name - servers 8.8.8.8,8.8.4.4;
}
```



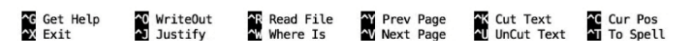


图 20-4　在/etc/dhcp/dhcpd.conf 最后添加设置

STEP4：设置/etc/default/isc-dhcp-server 的配置文件

需要指定通过 USB WiFi Dongle 来分享网络，所以请输入以下命令打开配置文件，如图 20-5 所示。

```
$ sudo nano /etc/default/isc - dhcp - server
```

把

```
INTERFACES = ""
```

改成为

```
INTERFACES = "wlan0"
```

STEP5：调整 wlan0 网卡的设置

通过文本编辑器修改网卡的设置，如图 20-6 所示。

```
$ sudo nano /etc/network/interfaces
```

添加＃注释掉原本的 wlan0 设置，然后添加以下设置，意思是指定这个 WiFi 网卡的地址为固定的 IP：192.168.42.1。

```
iface wlan0 inet static
address 192.168.42.1
netmask 255.255.255.0
up iptables - restore < /etc/iptables.ipv4.nat
```

```
GNU nano 2.2.6          File: /etc/default/isc-dhcp-server          Modified

# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts

#
# This is a POSIX shell fragment
#

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="wlan0"

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

图 20-5 设置/etc/default/isc-dhcp-server 文件

```
GNU nano 2.2.6          File: /etc/network/interfaces          Modified

auto lo

iface lo inet loopback

iface eth0 inet static
address 192.168.0.111
gateway 192.168.0.1
netmask 255.255.255.0
broadcast 192.168.0.255

# allow-hotplug wlan0
# iface wlan0 inet manual
# wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
# iface default inet dhcp


iface wlan0 inet static
address 192.168.42.1
netmask 255.255.255.0

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

图 20-6 设置 etc/network/interfaces 文件

STEP6：重新指定 wlan0 WiFi 网卡的 IP 地址

输入以下命令重新指定 wlan0 WiFi 网卡的 IP 地址，或者把树莓派重新开机。

```
$ sudo ifconfig wlan0 192.168.42.1
```

完成后，输入网络查询命令做确认，如图 20-7 所示。

```
$ ifconfig
```

```
pi@raspberrypi ~ $ sudo ifconfig wlan0 192.168.42.1
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:76:58:3e
          inet addr:192.168.0.111  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9417 errors:0 dropped:1 overruns:0 frame:0
          TX packets:3127 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8824616 (8.4 MiB)  TX bytes:361963 (353.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 80:1f:02:d3:8a:40
          inet addr:192.168.42.1  Bcast:192.168.42.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3282 errors:0 dropped:2 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:892231 (871.3 KiB)  TX bytes:1917 (1.8 KiB)
```

图 20-7　重新指定 wlan0 WiFi 网卡的 IP 地址

STEP7：设置/etc/hostapd/hostapd.conf 的配置文件

输入以下命令打开 hostapd 的配置文件。

```
$ sudo nano /etc/hostapd/hostapd.conf
```

修改内容如下，把 ssid 后的 Powenko_Pi_AP 改为树莓派迷你无线路由器名称，wpa_passphrase 后面的 raspberry 改为其密码，如图 20-8 所示。

```
interface = wlan0
driver = rtl871xdrv
ssid = Powenko_Pi_AP
hw_mode = g
channel = 6
macaddr_acl = 0
auth_algs = 1
ignore_broadcast_ssid = 0
wpa = 2
wpa_passphrase = raspberry
wpa_key_mgmt = WPA - PSK
wpa_pairwise = TKIP
rsn_pairwise = CCMP
```

STEP8：修改/etc/default/hostapd 的配置文件

打开文本编辑器，修改/etc/default/hostapd 的配置文件。

```
$ sudo nano /etc/default/hostapd
```

把原本的设置

```
#DAEMON_CONF = "" and edit it so it says
```



图 20-8　设置连接的 SSID 和密码

修改以下的设置,如图 20-9 所示。

```
DAEMON_CONF = "/etc/hostapd/hostapd.conf"
```

完成后按下 Ctrl+O 组合键存储,Ctrl+X 组合键离开文本编辑器 nano。

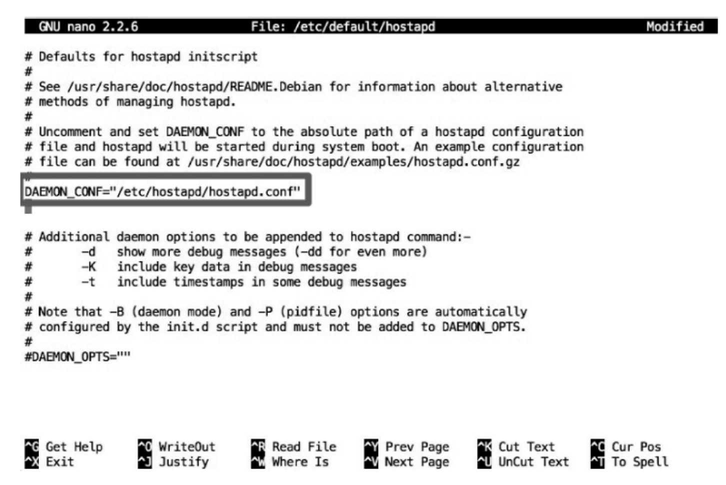


图 20-9　修改设置

STEP9：设置网络位置转点

要设置网络位置转点,打开文本编辑器修改/etc/sysctl.conf 的配置文件。

```
$ sudo nano /etc/sysctl.conf
```

在最后加上一行,如图 20-10 所示。

```
net.ipv4.ip_forward = 1
```

完成后按下 Ctrl+O 组合键存储,Ctrl+X 组合键离开文本编辑器 nano。

并且通过以下命令,重新开机。

```
$ sudo reboot
```



图 20-10　设置网络位置转点

引导后,执行以下命令,达到设置 IP 转送的功能。

```
$ sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
$ sudo sh -c "echo net.ipv4.ip_forward = 1 >> /etc/sysctl.conf"
```

STEP10:启用 NAT 的功能

输入以下命令,如图 20-11 所示,达到启用 NAT 的功能。

```
$ sudo iptables -F
$ sudo iptables -F -t nat
$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
$ sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED
-j ACCEPT
$ sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
$ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
$ sudo iptables-save > /etc/iptables.ipv4.nat
$ sudo iptables-restore < /etc/iptables.ipv4.nat
```

```
pi@raspberrypi ~ $ sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
pi@raspberrypi ~ $ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
pi@raspberrypi ~ $ sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED
-j ACCEPT
pi@raspberrypi ~ $ sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
pi@raspberrypi ~ $ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
pi@raspberrypi ~ $
```

图 20-11　启用 NAT 的功能

STEP11：测试配置文件是否正确

输入以下命令测试以上的配置文件是否正确，如图 20-12 所示。

```
$ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

```
pi@raspberrypi ~/8188/RTL8188C_8192C_USB_linux_v4.0.2_9000.20130911/wireless_tools/wireless_tools.
30.rtl $ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
drv->ifindex=3
l2_sock_recv==l2_sock_xmit=0x0xb94638
+rtl871x_sta_deauth_ops, ff:ff:ff:ff:ff:ff is deauth, reason=2
rtl871x_set_key_ops
rtl871x_set_key_ops
rtl871x_set_key_ops
rtl871x_set_key_ops
Using interface wlan0 with hwaddr 00:13:13:00:0f:97 and ssid 'Powenko_Pi_AP'
```

图 20-12　成功的画面

会出现驱动程序有问题的情况，如图 20-13 所示。

```
pi@raspberrypi ~ $ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
Line 2: invalid/unknown driver 'rtl871xdrv'
1 errors found in configuration file '/etc/hostapd/hostapd.conf'
```

图 20-13　驱动程序错误发生

STEP12：确认驱动程序名称

输入以下命令确认 USB WiFi 的型号，以本书为例是 RealTek RTL8188CUS 802.11n，如图 20-14 所示。

```
$ lsusb
```

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 8301:0089
Bus 001 Device 005: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter
```

图 20-14　确认 USB WiFi 的型号

STEP13：下载驱动源程序

下载驱动程序解压缩后，移动到 hostpad 的源程序路径下，如图 20-15 所示。

```
$ wget ftp://WebUser2:r3iZ6vJI@23.251.207.30/cn/wlan/RTL8188C_8192C_USB_linux_v4.0.2_
9000.20130911.zip
$ unzip RTL8188C_8192C_USB_linux_v4.0.2_9000.20130911.zip
```

```
$ cd RTL8188C_8192C_USB_linux_v4.0.2_9000.20130911/
$ cd wpa_supplicant_hostapd/
$ tar zxvf wpa_supplicant_hostapd-0.8_rtw_r7475.20130812.tar.gz
$ cd wpa_supplicant_hostapd-0.8_rtw_r7475.20130812/hostpad
```

```
plicant_hostapd-0.8_rtw_r7475.20130812 $ cd hostapd/
pi@raspberrypi ~/8188/RTL8188C_8192C_USB_linux_v4.0.2_9000.20130911/wpa_supplicant_hostapd/wpa_sup
plicant_hostapd-0.8_rtw_r7475.20130812/hostapd $ ls
Android.mk         dump_state.h             hostapd_cli.c             main.c
bsd_hostapd.conf   eap_register.c           hostapd.conf              Makefile
ChangeLog          eap_register.h           hostapd.deny              nt_password_hash.c
config_file.c      eap_testing.txt          hostapd.eap_user          README
config_file.h      hlr_auc_gw.c             hostapd.radius_clients    README-WPS
ctrl_iface.c       hlr_auc_gw.milenage_db   hostapd.sim_db            src
ctrl_iface.h       hostapd.8                hostapd.vlan              wired.conf
defconfig          hostapd.accept           hostapd.wpa_psk
dump_state.c       hostapd_cli.1            logwatch
```

图 20-15 下载 hostpad 驱动源程序

STEP14：删除原本 hostapd 主程序，但保留设置

先删除原本的主程序，但保留相关的设置信息。

```
$ sudo apt-get -y remove hostapd
```

STEP15：编译程序

进行程序编译，如图 20-16 所示。

```
$ make clean
$ make
$ sudo make install
```

```
pi@raspberrypi ~/8188/RTL8188C_8192C_USB_linux_v4.0.2_9000.20130911/wpa_supplicant_hostapd/wpa_sup
plicant_hostapd-0.8_rtw_r7475.20130812/hostapd $ make
  CC  main.c
  CC  config_file.c
  CC  ../src/ap/hostapd.c
../src/ap/wpa_auth_glue.c: In function 'hostapd_wpa_auth_send_ether':
../src/ap/wpa_auth_glue.c:351:6: warning: variable 'ret' set but not used [-Wunused-but-set-variab
```

图 20-16 编译程序

STEP16：解压缩 wireless_tools 源程序

解压缩 wireless_tools 源程序，如图 20-17 所示。

```
$ sudo cp /usr/local/bin/hostapd* /usr/sbin
$ cd ../../../wireless_tools
$ tar zxvf wireless_tools.30.rtl.tar.gz
```

```
plicant_hostapd-0.8_rtw_r7475.20130812/hostapd $  cd ../../../wireless_tools
pi@raspberrypi ~/8188/RTL8188C_8192C_USB_linux_v4.0.2_9000.20130911/wireless_tools $  tar zxvf wir
eless_tools.30.rtl.tar.gz
```

图 20-17 解压缩 wireless_tools 源程序

STEP17：编译 wireless_tools 源程序

编译和安装 wireless_tools 源程序，如图 20-18 所示。

```
$ cd wireless_tools.30.rtl
$ make clean
$ make
$ sudo make install
```

```
pi@raspberrypi ~/8188/RTL8188C_8192C_USB_linux_v4.0.2_9000.20130911/wireless_tools/wireless_tools.
30.rtl $ sudo make install
install -m 755 -d /usr/local/lib
install -m 644 libiw.a /usr/local/lib
install -m 755 -d /usr/local/sbin
install -m 755 iwconfig iwlist iwpriv iwspy iwgetid iwevent ifrename /usr/local/sbin
```

图 20-18　编译 wireless_toolsI 源程序

输入以下命令测试以上的配置文件是否正确。

```
$ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

STEP18：手机测试路由器

用手机或其他设备测试树莓派路由器是否可以连接上。但连接上后，却不能打开网页得到网络的内容，如图 20-19 所示。

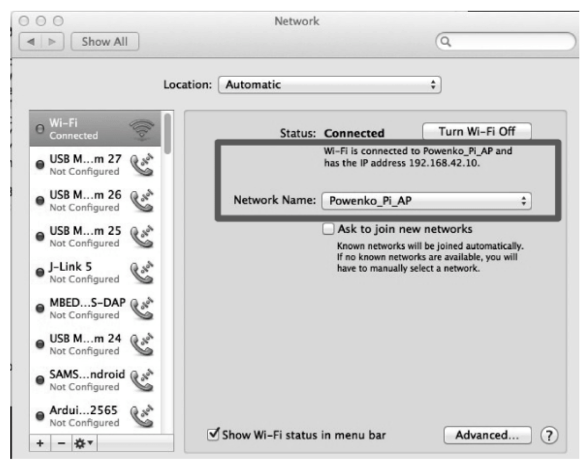


图 20-19　测试连接的热点

STEP19：更新 Hostpd 程序

因为 apt-get 版本有些问题，所以有人把修改后的放在网络上，输入以下命令下载调整后的程序，如图 20-20 所示。

```
$ wget http://www.adafruit.com/downloads/adafruit_hostapd.zip
```

```
pi@raspberrypi ~ $ wget http://www.adafruit.com/downloads/adafruit_hostapd.zip
--2014-08-29 18:21:09--  http://www.adafruit.com/downloads/adafruit_hostapd.zip
Resolving www.adafruit.com (www.adafruit.com)... 207.58.139.247
Connecting to www.adafruit.com (www.adafruit.com)|207.58.139.247|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 709582 (693K) [application/zip]
Saving to: `adafruit_hostapd.zip.1'

100%[=========================================================>] 709,582      697K/s   in 1.0s

2014-08-29 18:21:10 (697 KB/s) - 'adafruit_hostapd.zip.1' saved [709582/709582]
```

图 20-20　更新 Hostpd 程序

解压缩后，把需要的文件复制到系统中，并且修改用户权限。

```
$ unzip adafruit_hostapd.zip
$ sudo mv /usr/sbin/hostapd /usr/sbin/hostapd.ORIG
$ sudo mv hostapd /usr/sbin
$ sudo chmod 755 /usr/sbin/hostapd
```

6. 结果

输入以下命令测试配置文件是否正确

```
$ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

通过手机顺利连接并取得网络的数据，如图 20-21 所示。

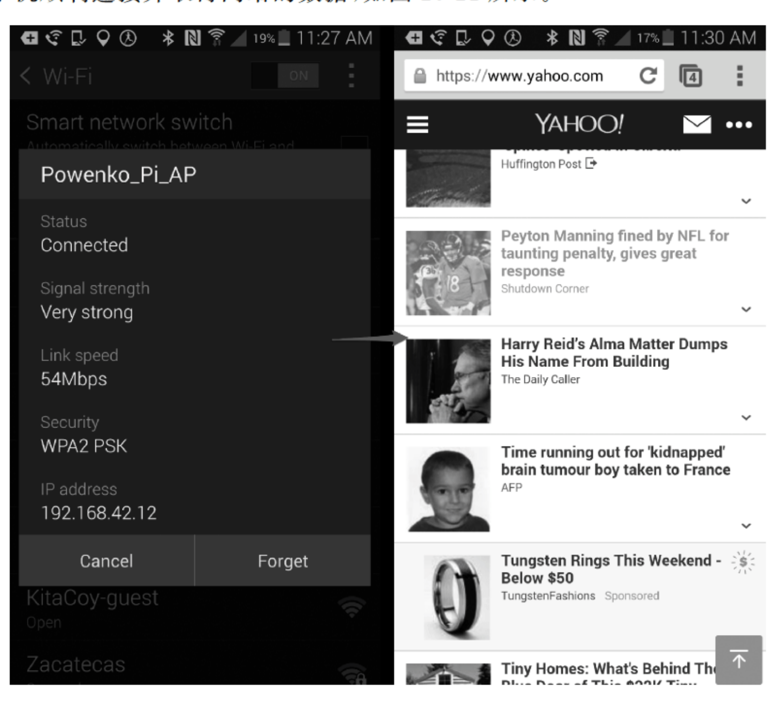


图 20-21　执行结果

7. 教学影片

完整的教学视频可以看 RaspberryPi_WiFiAP；如果指定驱动程序有问题，可以看视频 RaspberryPi_WiFiAP2_setupDriver RaspberryPi_WiFiAP3_final。

注意 如果想要开机时，自动引导路由器 hostapd 的功能，输入以下命令设置。

```
$ sudo update - rc.d hostapd defaults
```

输入以下命令删除设置。

```
$ sudo update - rc.d  - f hostapd remove
```

20.2 树莓派迷你网桥

案例 106

1. 介绍

本案例介绍如何把树莓派变成一个迷你无线网桥，这和案例 105 的路由器有什么不同呢？有两种方式可以处理网络封包的传递，一是桥接模式，二是路由模式。桥接模式就是单纯的把 WiFi 传入的网络封包直接转到以太网送出，IP 地址是向主机取得另外一个新的 IP 地址，而路由模式必须自行处理 DHCP、NAT、IP Forwarding 等服务。

2. 设备

同案例 105。

3. 接线

直接把网线接在树莓派上，把 USB WiFi Dongle 插到 USB 插槽上，并且确认网线可以正常工作。

4. 步骤

STEP1：更新安装软件

输入以下命令，更新安装软件。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

STEP2：安装网络桥接功能组件 bridge-utils

输入以下命令，安装网络桥接功能组件，如图 20-22 所示。

```
$ sudo apt - get  - y install bridge - utils
$ sudo apt - get install iw dnsmasq
$ sudo apt - get  - y install udhcpd
```

```
pi@raspberrypi ~ $ sudo apt-get -y install bridge-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  bridge-utils
0 upgraded, 1 newly installed, 0 to remove and 12 not upgraded.
```

图 20-22　安装 bridge-utils 软件

STEP3：修改无线网卡的设置值，加入桥接功能

修改无线网卡的设置值，加入桥接功能。

打开文本编辑器 nano，修改 /etc/network/interfaces 的配置文件。

```
$ sudo nano /etc/network/interfaces
```

主要是修改无线网成为桥接 br0。

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp

auto br0
iface br0 inet dhcp
  pre-up ip link set eth0 down
  pre-up ip link set wlan0 down
  pre-up brctl addbr br0
  pre-up brctl addif br0 eth0 wlan0
  pre-up ip addr flush dev eth0
  pre-up ip addr flush dev wlan0
  post-down ip link set eth0 down
  post-down ip link set wlan0 down
  post-down ip link set br0 down
  post-down brctl delif br0 eth0 wlan0
  post-down brctl delbr br0
```

完成后按下 Ctrl+O 组合键存储，Ctrl+X 组合键离开文本编辑器 nano。

STEP4：修改 hostapd. conf

修改 hostapd. conf 的设置。

```
$ sudo nano /etc/hostapd/hostapd.conf
```

修改内容如下，把 ssid 后的 Powenko_Pi_AP 改为树莓派迷你无线路由器名称，wpa_passphrase 后面的 raspberry 改为其密码，如图 20-23 所示。

```
interface=wlan0
driver=rtl871xdrv
```

```
bridge = br0
ssid = Powenko_Pi_bridge_AP
hw_mode = g
channel = 6
macaddr_acl = 0
auth_algs = 1
ignore_broadcast_ssid = 0
wpa = 2
wpa_passphrase = raspberry
wpa_key_mgmt = WPA - PSK
wpa_pairwise = TKIP
rsn_pairwise = CCMP
```

注意,这次加上了 bridge=br0。



图 20-23　修改后的 hostapd. conf 配置文件

STEP5：修改 sysctl. conf

打开文本编辑器,修改 sysctl. conf 的配置文件。

```
sudo nano /etc/sysctl.conf
```

把

```
# net.ipv4.ip_forward = 1
```

改为如图 20-24 所示。

```
net.ipv4.ip_forward = 1
```

完成后按下 Ctrl+O 组合键存储,Ctrl+X 组合键离开文本编辑器 nano。

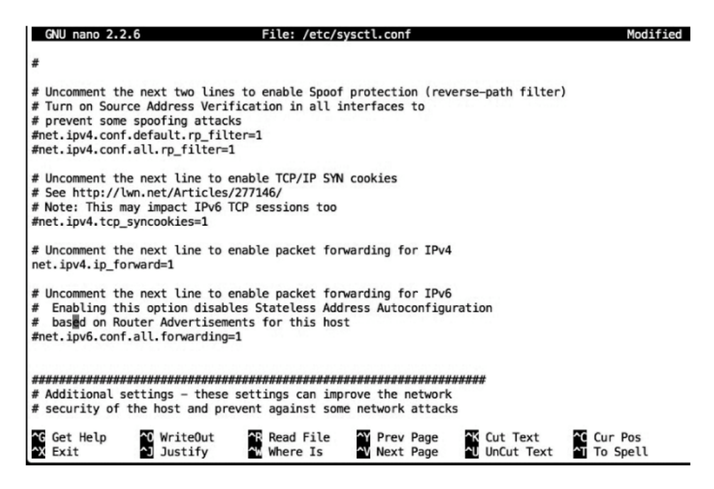


图 20-24　修改 sysctl.conf 的配置文件

STEP6:重新开机

重新开机,让网络设置组态重新生效。

```
$ sudo reboot
```

输入 ifconfig 确认是否有 br0 的虚拟设备,如图 20-25 所示。

```
$ ifconfig
```

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 8301:0089
Bus 001 Device 005: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter
pi@raspberrypi ~ $ ifconfig
br0       Link encap:Ethernet  HWaddr 00:13:13:00:0f:97
          inet addr:192.168.0.111  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::213:13ff:fe00:f97/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:451 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:84312 (82.3 KiB)  TX bytes:1332 (1.3 KiB)

eth0      Link encap:Ethernet  HWaddr b8:27:eb:76:58:3e
          inet addr:192.168.0.111  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::ba27:ebff:fe76:583e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8662 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5254 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10328156 (9.8 MiB)  TX bytes:2054064 (1.9 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:66 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6270 (6.1 KiB)  TX bytes:6270 (6.1 KiB)

wlan0     Link encap:Ethernet  HWaddr 00:13:13:00:0f:97
          inet6 addr: fe80::213:13ff:fe00:f97/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5613 errors:0 dropped:20 overruns:0 frame:0
          TX packets:8321 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2192160 (2.0 MiB)  TX bytes:10597661 (10.1 MiB)
```

图 20-25　出现 br0 的虚拟设备

5. 结果

重新打开后,树莓派迷你无线网桥的功能就会自动引导,可以通过手机顺利连接并取得网络的数据。注意一下手机上获取的 IP 地址,这个网络地址是按照现有的网络路由器,向它发出请求后取得的新 IP 地址,如图 20-26 所示。

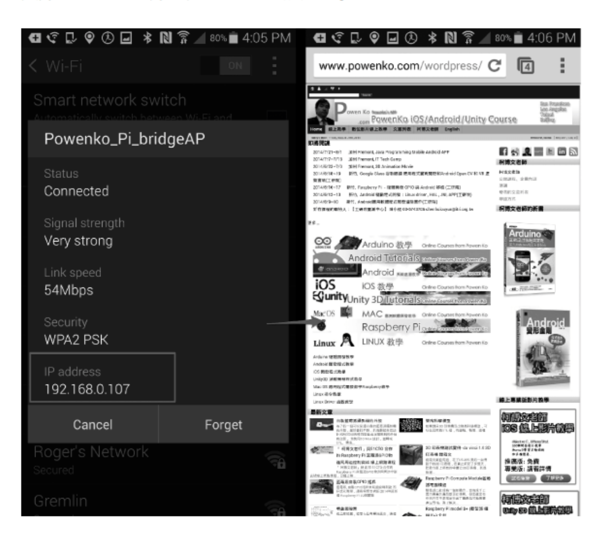


图 20-26　执行结果

6. 教学影片

完整的教学视频可以看 RaspberryPi_WiFi_Bridge。

第 21 章 树莓派机器人

2014 年巴西主办的世足赛是不是看不过瘾的？结束之后是不是有一种失落感，想继续看足球赛？你支持的足球队伍是否赢得 2014 年世界杯足球赛的冠军？如果不是德国队也没关系，现在还有另外一个世界杯足球赛正在举办，那就是 Robocup 2014 机器人的世界杯足球赛，如图 21-1 所示。

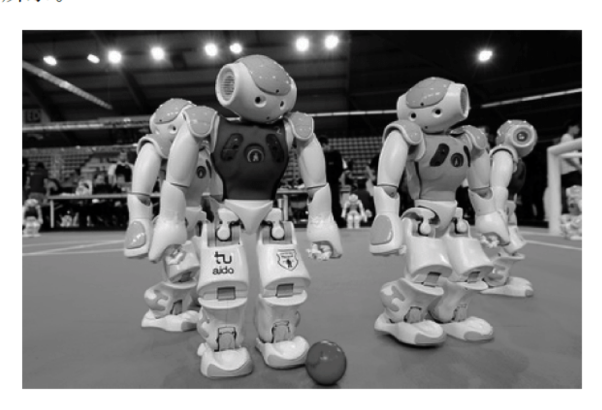


图 21-1 Robocup 2014 机器人的世界杯足球赛

有可能自己做一个树莓派机器人吗？当然可能，本章介绍如何创建出自己专用的树莓派机器人。首先研究机器人为何可以移动，最主要的零件是 SERVO 步进电动机，这种电动机跟前面树莓派小汽车所介绍的直流电动机，最大的不同是可以控制旋转的角度，但是在介绍步进电动机之前，必须要先介绍控制 SERVO 步进电动机的方法，就是 PWM 的输出，这是控制 SERVO 步进电动机旋转角度的方法。

21.1 树莓派 SOS 求救灯

案例 107

1. 目的

本案例利用 LED 灯的显示做一个求救灯 SOS 的信号出去，快速地闪烁 3 次，再慢速的

闪烁 3 次，最后快速地闪烁 3 次。通过自己写的 Python 程序控制 GPIO 的 PWM 函数来达到目的。

2. 设备

(1) 树莓派开发板；

(2) 2 个 10kΩ 的电阻(棕黑橘)；

(3) 2 个 LED 灯；

(4) 面包板；

(5) 电线若干。

3. 接线

把树莓派关机，拔掉电源，按照表 21-1 和图 21-2 的硬件接线图，重新连接硬件线路。

表 21-1　硬件引脚

树莓派引脚 pin	组 件 引 脚
Pin 7 / GPIO4	第一个 LED 灯的长脚
Pin 6/ GND	第一个 LED 灯的短脚，先接 10kΩ 的电阻，再接到树莓派

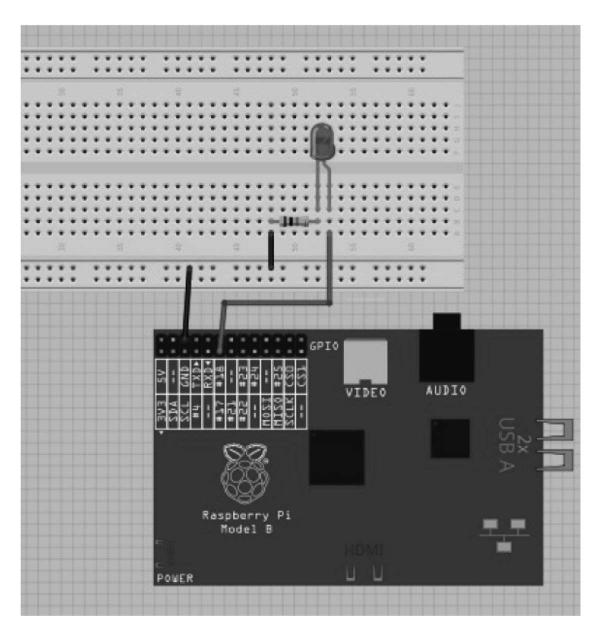


图 21-2　实际硬件接线

4. 步骤

打开文本编辑器，创建 sos.py 文件。

```
$ nano sos.py
```

把以下程序复制粘贴上，通过 Ctrl+O 组合键存储，Ctrl+X 组合键离开程序。

样例程序：sos. py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
3.  import time,RPi.GPIO as GPIO
4.
5.  GPIO.setmode(GPIO.BCM)
6.  GPIO.setup(17,GPIO.OUT)
7.  p1 = GPIO.PWM(17,1)            # channel = 17 frequency = 1Hz
8.  p1.start(0)
9.
10. p1.ChangeDutyCycle(50)
11. p1.ChangeFrequency(1)          # frequency = 1Hz
12. time.sleep(3);
13. p1.ChangeFrequency(1 * 2)      # frequency = 2Hz
14. time.sleep(1.5);
15. p1.ChangeFrequency(1)          # frequency = 1Hz
16. time.sleep(3);
17.
18. p1.stop()
19. GPIO.cleanup()
```

程序解说

第 1 行：执行环境的定义文档，Python 程序存放路径。

第 2 行：注释的写法，前面加个 # 就成为注释。

第 5 行：定义 GPIO. BCM。

第 8 行：设置 GPIO 17/ Pin 11 的引脚，负责数据输出 GPIO. OUT。

第 10 行：设置为 50，意思是在一定的时间内 50%亮 50%暗。

第 11 行：调整 GPIO17 的频率。

执行情况

这里 PWM 设置为 50%，所以在 1Hz 的情况下，1 秒钟有 0.5 秒亮和 0.5 秒暗。

执行此 Python 程序，出现以下情况：

(1) 前 3 秒钟，有 3 次 0.5 秒亮和 0.5 秒暗。

(2) 在接下来的 1.5 秒钟，完成 3 次 0.25 秒亮和 0.25 秒暗。

(3) 在最后 1.5 秒钟，完成 3 次 0.5 秒亮和 0.5 秒暗。

通过计算调整 LED 的灯光时间达到 SOS 信号的效果，快速地闪烁 3 次、再慢速的闪烁 3 次、最后快速地闪烁 3 次，如图 21-3 所示。

```
$ sudo python sos.py
```

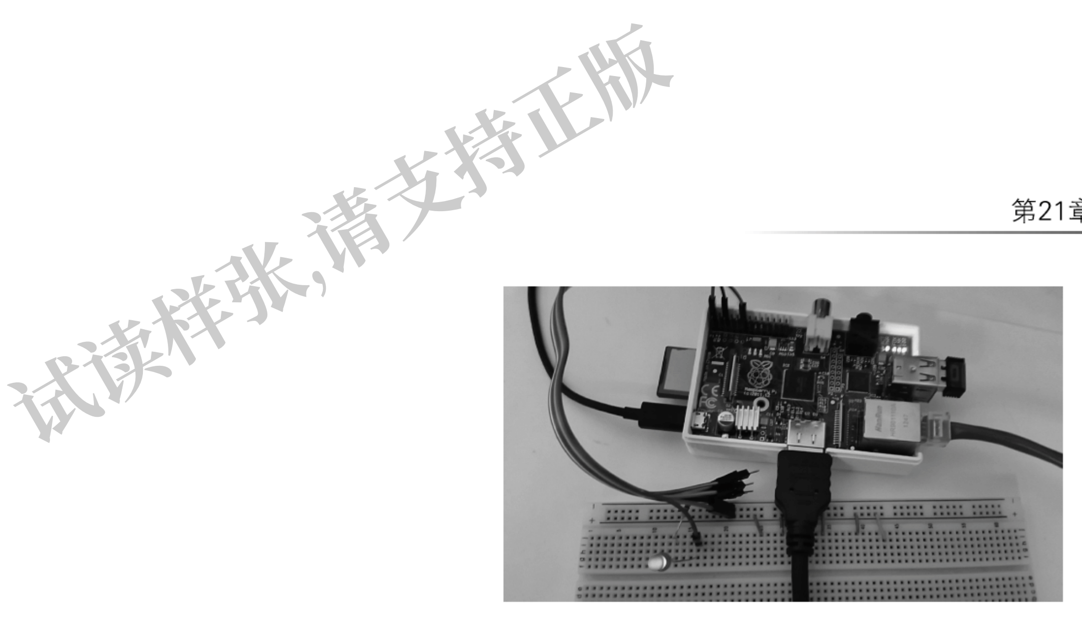


图 21-3　执行结果

5. 教学影片

完整的教学视频可以看 RaspberryPi_PWM_SOS。

21.2　SERVO 步进电动机

案例 108

1. 介绍

SERVO 步进电动机是非常重要的一个组件,如图 21-4 所示,SERVO 步进电动机的特别之处是可以控制电动机的旋转角度。步进电动机是脉冲电动机的一种,具有如齿轮状突起(小齿)相契合的定子和转子,通过切换流向定子线圈中的电流,以一定角度逐步转动。步进电动机的特征是采用开环(Open Loop)控制方式处理,不需要传感器(Sensor)或编码器(Encoder);因为切换电流触发器的是脉波信号,所以不需要位置检测和速度检测的传感器。因此,步进电动机可准确地按比例随脉波信号而转动,实现精确地控制位置和速度,且稳定性好。



图 21-4　TOWER PRO MG996R SERVO 步进电动机

当前有各种各样的 SERVO 步进电动机,表 21-2 是常见的步进电动机。

表 21-2 常见的 SERVO 步进电动机

名　　称	扭力	扭矩	速度
TOWER PRO MG996R	11kg	9.4kg/cm(4.8V),11kg/cm (6V)	0.17s/60°(4.8V) 0.14s/60° (6V)
TOWER PRO 辉盛 MG995	13kg	13kg/cm(6V)	0.17s/60°(4.8V) 0.13s/60°
TOWER PRO MG90S	2.8kg	2.4kg/cm(4.8V) 2.8kg/cm (6V)	0.11s/60°(4.8V) 0.09s/60° (6.0V)
TOWER PRO MG945	13kg	10.5kg/cm(4.8V) 10.6kg/cm (6V)	0.20s/60°(4.8V) 0.17s/60° (6.0V)
TOWER PRO MG946R	13kg	10.5kg/cm(4.8V),13kg/cm (6V)	0.20s/60°(4.8V) 0.17s/60° (0V)

在电动机系统中,最常用的电动机是步进电动机和伺服电动机,其中步进电动机主要分为 2 相、5 相、微步进系统;伺服电动机主要分为 DC 伺服和 AC 伺服。RC SERVO 为多数规模机器人最主要的制动器,它体积小、重量轻并可提供精确的旋转角度与足够的扭力,国产品牌有祥仪、广营、栗研等,日韩系有 KONDO、HITEC,RC SERVO 大部分是通过 PWM (脉冲宽度调制)来控制。

RC SERVO 控制界面可分为以下几种:PWM、RS232、RS485、I2C,其中 PWM 控制是一种受欢迎的控制方式,本案例也采用 PWM 来控制 SERVO 步进电动机。一般 SERVO 步进电动机是利用 duty cycle high 的宽度来控制 RC SERVO 的旋转角度,例如 KONO KRS-788HV 所接受的 duty cycle high 宽度介于 700μs～2300μs 之间,因此用户必须提供此范围的 PWM 信号,如图 21-5 所示。

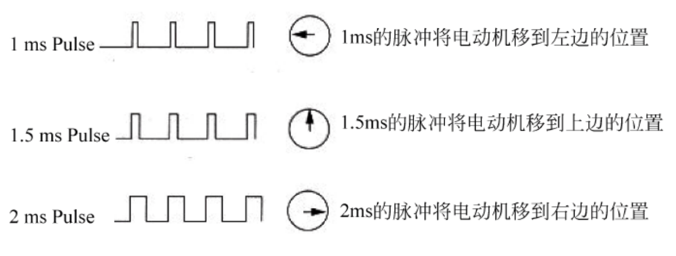


图 21-5 改变 duty cycle high 宽度来控制 RC SERVO 角度

以 KRS-788HV 为例,给予 1500μs 的 PWM 信号,它会转到 90°处(因为 1500μs 宽度为中间值,KRS-788HV 的可动角度为 0°～180°,所以对应到中间位置)。改变 PWM duty cycle 来控制 RC SERVO 角度,不同的 RC SERVO 生产商有不同的默认 PWM 与旋转角度范围(一般是 180°～270°),所以在使用前,先仔细阅读 SERVO 步进电动机使用手册。有的有角度的限制,有的转速的设置都不一样,挑选 SERVO 步进电动机重点要看扭力和转速,以免扭力太小不能顺利旋转。

2. 应用范围

遥控汽车；

机械手臂；

机器人。

3. 旋转角度

通过控制 SERVO 步进电动机旋转角度看树莓派如何与 SERVO 步进电动机做交互。让 SERVO 步进电动机由 0°,一次转 1°,旋转到 180°,再由 180°,一次少一度,旋转到 0°。

4. 设备

(1) 树莓派开发板；

(2) 1 个 SERVO 步进电动机；

(3) 面包板；

(4) 接线若干。

5. 硬件规范

本案例用 TOWER PRO MG996R 金属齿轮,11kg 大扭力服务器只要 6V 电压就可以转动,是 MG995 升级产品,是当前性价比最高的大扭力舵机之一。与 MG946R 相比,MG996R 速度快,但拉力稍小,如图 21-6 所示。

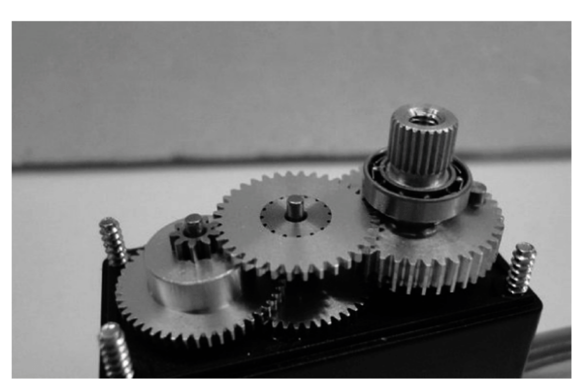


图 21-6　SERVO 步进电动机的内部构造

厂家编号：MG996R；

产品净重：55g；

产品尺寸：40.7mm×19.7mm×42.9mm；

产品拉力：9.4kg/cm(4.8V),11kg/cm(6V)；

反应速度：0.17sec/60degree(4.8V) 0.14sec/60degree(6V)；

工作电压：4.8～7.2V；

工作温度：0～55℃；

齿轮形式：金属齿轮；

工作区：5μs。

一般以 PWM 信号控制的 RC SERVO 对外连接线如图 21-7 所示，白线为 PWM 信号线，红线为电源线，黑线为地线，用户通常通过微处理器连接 SERVO 控制器来提供 PWM 信号及电源。

此设备有 3 个引脚，如图 21-7 所示。

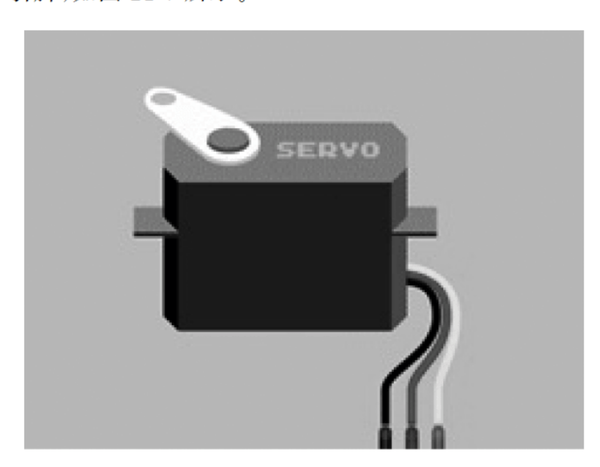


图 21-7　SERVO 步进电动机的接线

（1）VCC 电源；

（2）GND 接地；

（3）Pulse 脉冲（用来控制角度）。

按照经验，当通过树莓派连接 SERVO 电动机，电动机的扭力就会不够，并且会让树莓派不正常的突然关机，这是因为电动机用电量比较大，解决方法如下：

电动机的 VCC 电源额外接上电压 5V 和电流 2A 以上的 USB 充电器，如图 21-8 所示，并把 GND 接地接到树莓派上。



图 21-8　供电器

6. 接线

参考表 21-3 和图 21-9 接线图，把硬件连接起来。

表 21-3 引脚

树莓派引脚	组 件 引 脚	外 部 电 源
Pin 11/GPIO17	SERVO 步进电动机的 Pulse 脉冲	无
GND	SERVO 步进电动机的 GND 接地	GND 接地
无	SERVO 步进电动机的 VCC 电源	VCC 电源

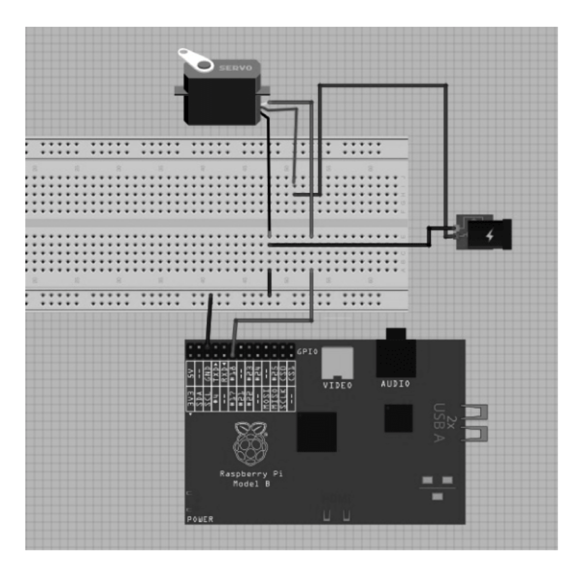


图 21-9 接线图

样例程序：sample\ch12\pwm. py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
3.  import time,RPi.GPIO as GPIO
4.
5.  GPIO.setmode(GPIO.BCM)
6.  GPIO.setup(17,GPIO.OUT)
7.  p1 = GPIO.PWM(17,100)    # channel = 17 frequency = 100Hz
8.  p1.start(0)
9.
10. for dc in range (0,100,5):
11.     p1.ChangeDutyCycle(dc)
12.     time.sleep(1);
13.
14. p1.stop()
15. GPIO.cleanup()
```

程序解说

第3行：引入GPIO的函数库。

第7行：设置PWM为GPIO17/引脚11。

第10行：使用循环由0到100,每次调整5。

第11行：设置SERVO步进电动机移动电动机旋转角度。

7. 结果

不同的SERVO MOTOR会因为PWM的不同,造成SERVO电动机旋转角度不同,请按照实际情况调整PWM的范围值,确认旋转的角度,如图21-10所示。

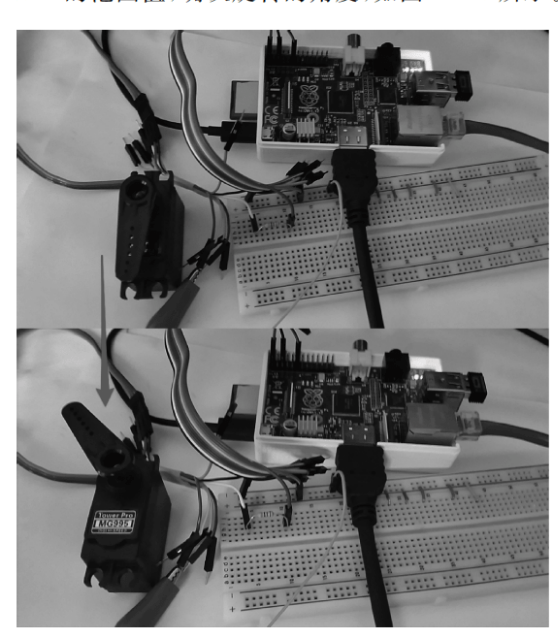


图21-10 执行结果

8. 教学影片

完整的教学视频可以看RaspberryPi_PWM_ServoMotor.mov。

21.3 控制机器人手臂角度

案例109

1. 介绍

本案例介绍如何做出一个机器人。市面上贩卖的机器人,产品介绍中都会提到这个机器拥有多少关节,仔细看一下商品的照片,如图21-11所示,这些关节就是案例108提到的步进电动机所组成的。

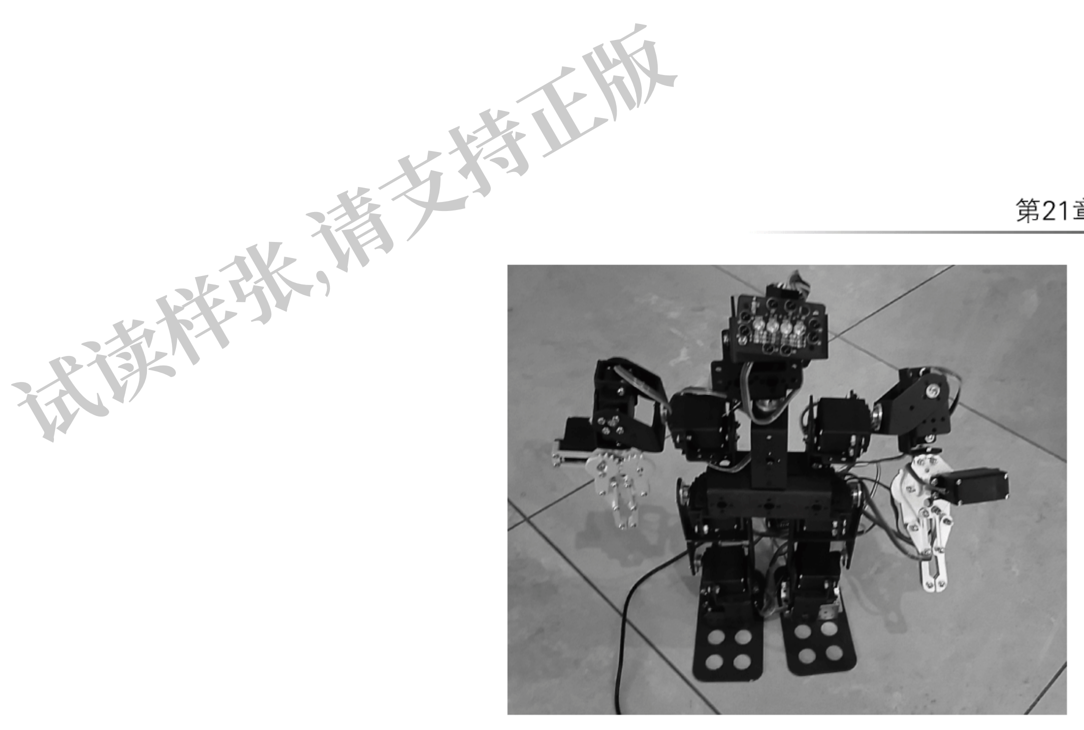


图 21-11　作者自制机器人

如果要控制机器人的所有关节(最少 16 个关节),程序和硬件线路就过于复杂,所以在本案例中用键盘控制机器人的两个关节,这样就能达到手臂的移动,机器人左右手的步进电动机可以实现举起左右手,希望大家知道原理后,扩展到所有的关节。

2. 设备

(1) 树莓派开发板;

(2) 2 个 SERVO 步进电动机;

(3) 面包板;

(4) 接线若干;

(5) 机器人零件(选配)。

3. 硬件规范

此设备有三个接脚,分别是:

(1) VCC 电源;

(2) GND 接地;

(3) Pulse 脉冲(用来控制角度用)。

4. 接线

参考表 21-4 和图 21-12 接线图,把硬件连接起来。

表 21-4　引脚

树莓派引脚	组件引脚	外部电源
Pin 11/GPIO17	SERVO 步进电动机 1 的 Pulse 脉冲	无
Pin 12/GPIO18	SERVO 步进电动机 2 的 Pulse 脉冲	无
GND	SERVO1 和 SERVO2 步进电动机的 GND 接地	GND 接地
无	SERVO1 和 SERVO2 步进电动机的 VCC 电源	VCC 电源

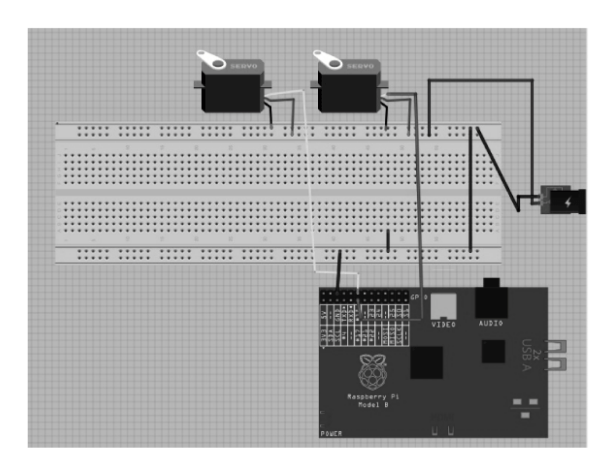


图 21-12　接线图

样例程序：sample\ch12\robot1.py

```
1.  #!/usr/bin/env
2.  #__author__ = "Powen Ko"
3.  import RPi.GPIO as GPIO
4.  from time import sleep
5.  import threading
6.  import sys
7.  import sys,tty,termios
8.
9.  def getch():                                    # 键盘处理
10.     fd = sys.stdin.fileno()
11.     old = termios.tcgetattr(fd)
12.     try:
13.         tty.setraw(fd)
14.         return sys.stdin.read(1)
15.     finally:
16.         termios.tcsetattr(fd,termios.TCSADRAIN,old)
17.
18. class KeyEventThread(threading.Thread):         # 多执行序
19.     def run(self):
20.         print("thread");
21.         Fun()
22.
23. def Fun():
24.     print("Fun")
25.     while True:
26.         key=getch()                             # 按键处理
27.         print(key)
```

```
28.         if key == 'q':
29.             funExit()
30.             exit()
31.             return
32.         elif key == 'w':
33.             funSpeed(160, -1)
34.         elif key == 's':
35.             funSpeed(80, -1)
36.         elif key == 'x':
37.             funSpeed(10, -1)
38.         elif key == 'e':
39.             funSpeed(-1,100)
40.         elif key == 'd':
41.             funSpeed(-1,40)
42.         elif key == 'c':
43.             funSpeed(-1,15)
44.         else:
45.             print("key = " + key)
46.     time.sleep(0.5)
47.     return
48.
49.
50.
51. def funSpeed(i1,i2):
52.     if i1 > 0:
53.         duty = float(i1) / 10.0 + 2.5
54.         p1.ChangeDutyCycle(duty)
55.     if i2 >= 0:
56.         duty = float(i2) / 10.0 + 2.5
57.         p2.ChangeDutyCycle(duty)
58.
59.
60.
61. def funInit():
62.     GPIO.setmode(GPIO.BOARD)                # 设置 2 个输出的引脚
63.     GPIO.setup(Motor1,GPIO.OUT)
64.     GPIO.setup(Motor2,GPIO.OUT)
65.
66. def funExit():                              # 关闭电动机的动作
67.     print "Stopping motor"
68.     GPIO.cleanup()
69.
70.
71. Motor1 = 11                                 # 17
72. Motor2 = 12                                 # 18
73.
```

```
74.
75. print("Press 'q' to exit")
76. print("'w','s','x' = left arm,   'e','d','c' = right arm ")
77. funInit()
78. p1 = GPIO.PWM(Motor1,100)          # PWM 的频率为 100Hz
79. p1.start(0)
80. p2 = GPIO.PWM(Motor2,100)
81. p2.start(0)
82. funSpeed(20,20)
83.
84.
85. kethread = KeyEventThread()
86. kethread.start()
```

程序解说

第 9～16 行：获取用户的按键。

第 18～21 行：路复用。

第 61～64 行：设置 PWM 为 GPIO17/引脚 11 和 GPIO18/引脚 12。

第 51～57 行：设置 SERVO 步进电动机移动电动机旋转角度。

第 85～86 行：设置 KeyEventThread 函数为执行路复用。

5. 结果

输入以下命令引导程序，通过 w 键、s 键、x 键控制第一个步进电动机，通过 e 键、d 键、c 键控制第二个步进电动机，q 键离开程序，如图 21-13 所示。

```
pi@raspberrypi ~/code $ sudo python robot1.py
Press 'q' to exit
'w','s','x'=left arm,  'e','d','c'=right arm
thread
Fun
w
s
w
x
s
e
d
s
x
w
q
```

图 21-13　执行结果

```
$ sudo python robot1.py
```

可以看到机器人的关节移动，如图 21-14 所示。

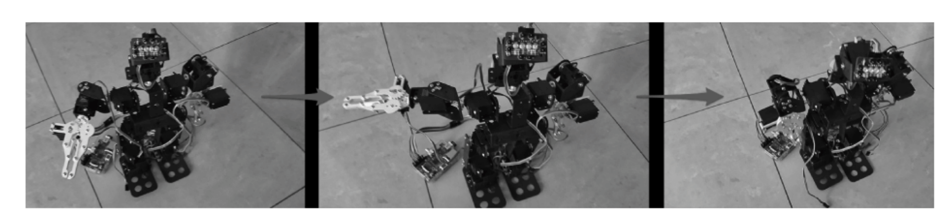


图 21-14　执行效果

6. 教学影片

完整的教学视频可以看 RaspberryPi_Robot1。

21.4　iOS/Android 手机遥控机器人

案例 110

使用相同的“19.7 控制小汽车”的方法，在 Android 系统安装 JuiceSSH——SSH Client。这款免费的 Android SSH APP，在 Android Store 上搜索 Juice SSH 就可以找到，使用 SSH 远程连接到树莓派机器人上，通过 WiFi 无线网络就能够控制。

如果想使用 iOS 智能手机，从远程 SSH 访问控制，可以使用 WebSSH 这款免费的 APP，使用起来相当便利好用。在 APP Store 上搜索 SSH 就可以找到。

把树莓派机器人的电源打开，连接和确认树莓派机器人已经顺利连上同一个局域网的 WiFi 无线网络，并且确认配置文件的网络地址是否正确。

通过 SSH 连接上之后，输入以下命令引导程序。

```
$ sudo python robot1.py
```

通过 w 键、s 键、x 键控制第一个步进电动机，通过 e 键、d 键、c 键控制第二个步进电动机，q 键离开程序，如图 21-15 所示。



图 21-15　通过 Android 希望控制树莓派机器人

21.5　3D 打印树莓派机器人外壳

案例 111

为了让大家更有成就感，推荐可以在拍卖网站上购买 Arduino 或树莓派的机器人组件，不过价格都好几百元甚至千元，如果有 3D 打印机也可以下载以下的 3D 模型，如图 21-16 所示，自己打印与组装。网址为 http://letsmakerobots.com/node/38881。



Attachment	Size
Gutsy-LowerParts.zip	688.43 KB
Gutsy-TorsoPart.zip	1.14 MB
Left-Arm-Lower-Part.zip	2.48 MB
Left-Arm-Upper-Part.zip	1.3 MB
Right-Arm-Lower-Part.zip	2.66 MB
Right-Arm-Upper-Part.zip	1.23 MB
Gutsy-CoinSlotLowerLeftArm.zip	2.61 MB
Gutsy-CoinSlotLowerRightArm.zip	2.68 MB
Gutsy-Head.zip	731.47 KB



图 21-16　机器人 3D 打印模型文件

第 22 章

树莓派屏幕与数字相框

有很多 TFT 的 LCD 的屏幕,但是专用于树莓派屏幕的并不多,本章把几个常见的 TFT 的 LCD 的屏幕,修改为符合 Raspberry Pi 的屏幕。但是做之前先想一下,当前有几种方法可以接上屏幕:

(1) 通过 Raspberry Pi 本身的 HDMI 输出。

(2) 通过 Raspberry Pi 本身的 RCA 输出,如图 22-1 所示。

(3) 通过转换器,把 HDMI 的接口转换成其他显示器接口,例如 VGA。

最简单的方法,就是使用带有 HDMI /RCA/VGA 接口的屏幕,直接接上就可以显示,最多就是设置屏幕的尺寸,打开 SD 卡根目录中的 config. txt 设置文件,调整屏幕显示模式。



图 22-1　HDMI 转 VGA 的转换器

22.1　TFT 类型的 LCD

案例 112

1. 介绍

有很多 TFT 的 LCD 的屏幕,作者手上有几个 SPI TFT LCD 屏幕,但是每个使用方法都不太相同,最主要是屏幕的驱动程序和芯片的差异。本案例介绍 Tontec 2014 2. 8 寸

TFT LCD,如图 22-2 所示,硬件规范如下:

(1) 240×320 RGB 像素 16bit,65K 颜色。

(2) 触控屏幕,直到本书出版时,驱动程序没有支持触控功能。

(3) 4 个按钮。

(4) CPLD 芯片(EPM3032)。

(5) (Texas Instruments) company TSC2003 chip。

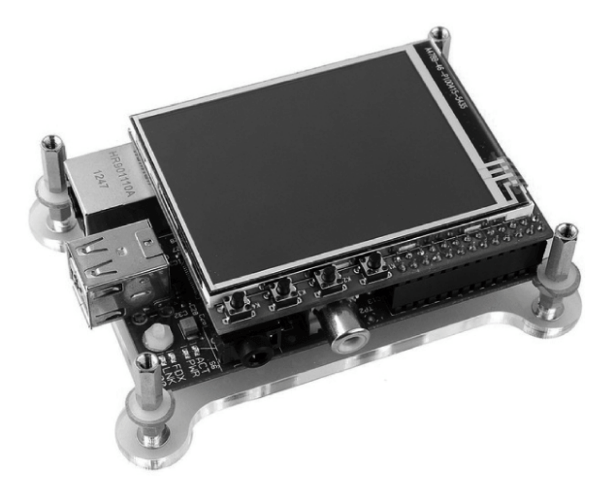


图 22-2 Tontec 2014 2.8 寸 TFT LCD 240×320 RGB 屏幕

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 1 个 Tontec 2.8 寸 TFT LCD;

(3) 面包板;

(4) 接线若干。

3. 接线

首先把树莓派关机,把 Tontec 2.8 寸 TFT LCD 专用 GPIO 接口,接在树莓派 GPIO 上。

4. 步骤

STEP1:更新安装软件

输入以下命令,更新安装软件。

```
$ sudo apt - get update
```

STEP2:下载驱动程序源程序

因为官方版本会占用树莓派 100%的 CPU,所以柯老师修改了一个版本,把 CPU 占用

率降到10%,放到 github 上的 https://github.com/powenko/Tontec-2.8-inch-TFT-LCD-240x320-RGB-RaspberryPi,如图22-3所示。输入以下命令,下载驱动程序源程序。

```
$ mkdir ~/download
$ cd ~/download
$ wget https://github.com/powenko/Tontec-2.8-inch-TFT-LCD-240x320-RGB-RaspberryPi/archive/master.zip
```

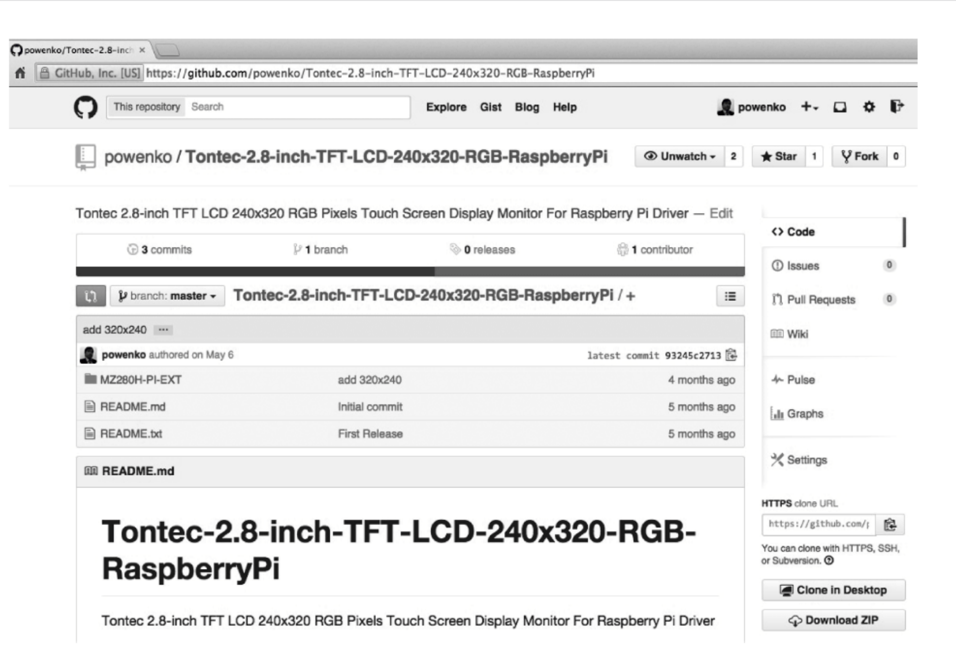


图22-3　驱动程序源程序下载

STEP3:解压缩

输入以下命令,解压缩驱动程序。

```
$ unzip master.zip
```

STEP4:确定分辨率

必须想好屏幕要设置的分辨率,如果是1∶1,就设置为320×240,但这个分辨率对X-Windows来说太小了,也可以设置为640×480,但这样对文字窗口的字又太小了。所以请按照实际需求来确定分辨率。

打开文本编辑器,修改/boot/config.txt的配置文件。

```
$ sudo nano /boot/config.txt
```

在文件最后添加 TFT LCD 的分辨率,如图 22-4 所示。

```
framebuffer_width = 320
framebuffer_height = 240
```

完成后,按下 Ctrl+O 组合键存储,Ctrl+X 组合键离开文本编辑器 nano。

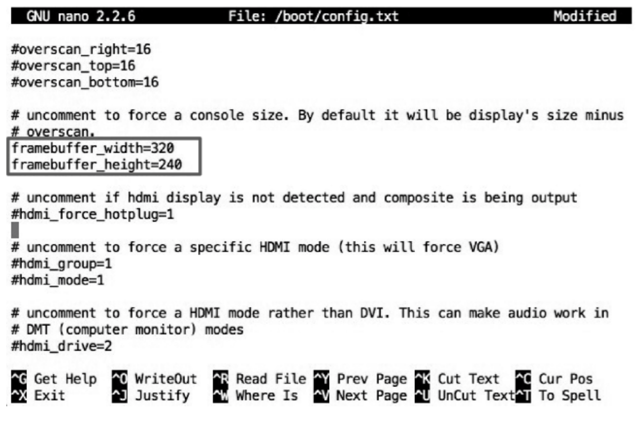


图 22-4 设置 config. txt

注意

/boot/config. txt 修改之后,会影响到树莓派的 AV 端子、HDMI 的分辨率,最后输出的分辨率应该都是一样。

STEP5:编译程序

输入以下命令,进行程序编译。

```
$ cd Tontec - 2.8 - inch - TFT - LCD - 240x320 - RGB - RaspberryPi - master
$ cd MZ280H - PI - EXT\src
```

这里有两个源程序,一个是 320×240 的 src_320x240 的路径,另外一个是 640×480 的 src_640x480 的路径,进入该路径,因为作者 config. txt 的设置是 320×240,所以执行程序如图 22-5 所示。

```
$ cd src_320x240
$ make
```

```
pi@raspberrypi ~/lcd/Tontec-2.8-inch-TFT-LCD-240x320-RGB-RaspberryPi-master/MZ28
0H-PI-EXT/src_320x240 $ make
gcc -o mz280h -l rt mz280h.c bcm2835.c
```

图 22-5 编译程序

STEP6：修改权限

修改编译出来的执行文件的权限，如图 22-6 所示。

```
$ sudo chmod +x mz280h
```

```
0H-PI-EXT/src_320x240 $ ls
bcm2835.c  bcm2835.h  Makefile  mz280h  mz280h.c  mztx28h
pi@raspberrypi ~/lcd/Tontec-2.8-inch-TFT-LCD-240x320-RGB-RaspberryPi-master/MZ28
0H-PI-EXT/src_320x240 $ sudo chmod +x mz280h
```

图 22-6　修改文件权限

STEP7：复制执行文件

复制执行文件到/usr/bin/，如图 22-7 所示。

```
$ sudo cp mz280h /usr/bin/
```

```
0H-PI-EXT/src_320x240 $ sudo cp mz280h /usr/bin
pi@raspberrypi ~/lcd/Tontec-2.8-inch-TFT-LCD-240x320-RGB-RaspberryPi-master/MZ28
```

图 22-7　复制执行文件

STEP8：设置开机自动执行

设置开机自动执行，通过文本编辑器 nano 创建文件。

```
$ sudo nano /etc/rc.local
```

在 exit 0 之前，写入以下内容，如图 22-8 所示，完成后，按下 Ctrl+O 组合键存储，Ctrl+X 组合键离开文本编辑器。

```
/usr/bin/mz280h &
```



图 22-8　修改/etc/rc.local 文件

5. 结果

输入以下命令重新开机就能看到效果,如图 22-9 所示。

```
$ sudo reboot
```

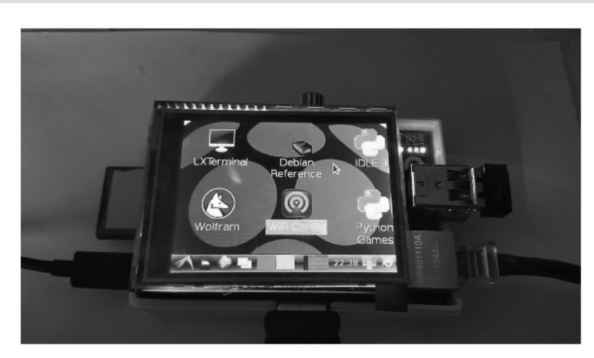


图 22-9　执行效果

6. 教学影片

完整的教学视频可以看 RaspberryPi_TFTLCD。

注意

查询 CPU 执行情况的命令如下:

```
$ ps aux | less
```

22.2　使用 SPI 的 TFT LCD

案例 113

1. 介绍

本案例介绍通过树莓派的 GPIO,使用 SPI 输送数据到 TFT LCD 的显示器上,并不是所有的 TFT LCD 都可以正常工作,当前确定有三个 TFT LCD 可以正常工作:

(1) Adafruit 1.8″ TFT LCD;

(2) SainSmart 1.8″ TFT;

(3) HY 1.8 SPI-1.8″ TFT128×160,如图 22-10 所示。

2. 设备

需要有以下硬件:

(1) 树莓派开发板;

(2) 1 个 Tontec 2.8 寸 TFT LCD;

(3) 面包板;

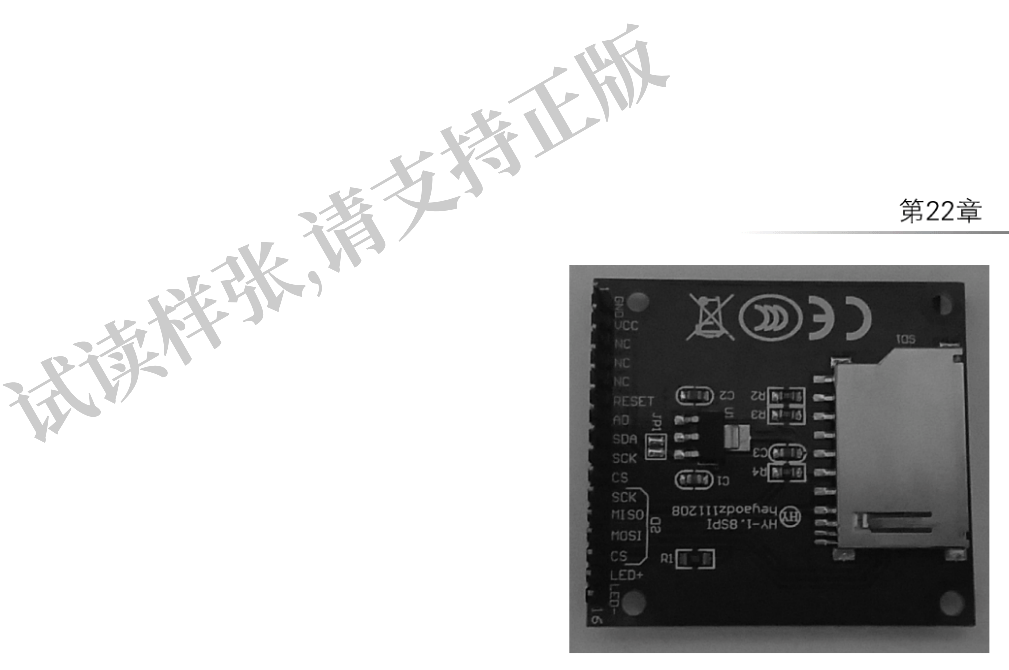


图 22-10　HY 1.8 SPI-1.8″ TFT128×160 外形

(4) 接线若干。

3. 接线

参考表 22-1 和图 22-11 SainSmart 1.8″ TFT 硬件线路图,并对照图 22-12 树莓派 GPIO 引脚号码对应图,把 TFT LCD 接到树莓派上。

表 22-1　引脚

树莓派的引脚	Adafruit 1.8″	SainSmart 1.8″ TFT	HY 1.8 SPI-1.8″ TFT128×160
5V /引脚 2	LTE	无	LED+
GND/引脚 6	无	无	LED−
GPIO 11/引脚 23	SCK	SCL	SCK
GPIO 10/引脚 19	MOSI	SDA	SDA
GPIO 8/引脚 24	TFT_CS	CS	CS
GPIO 24/引脚 18	D/C	RS/DC	A0
GPIO 25/引脚 22	RESET	RES	RESET
5V/引脚 4	VCC	VCC	VCC
GND/引脚 14	GND	GND	GND

4. 步骤

STEP1:更新安装软件

输入以下命令,更新安装软件。

```
$ sudo apt - get update
```

STEP2:下载对应固件

输入以下命令,下载对应固件,如图 22-13 所示。

```
$  sudo wget http://goo.gl/1BOfJ - O /usr/bin/rpi - update
```

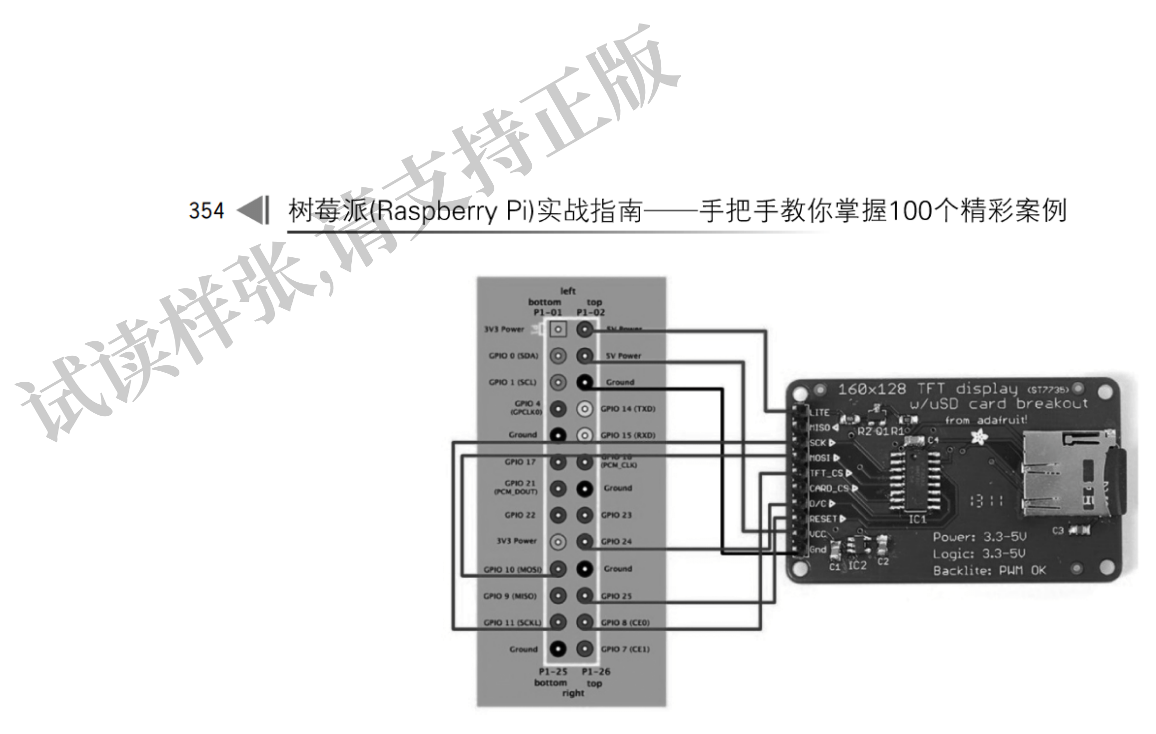


图 22-11 SainSmart 1.8" TFT 硬件线路

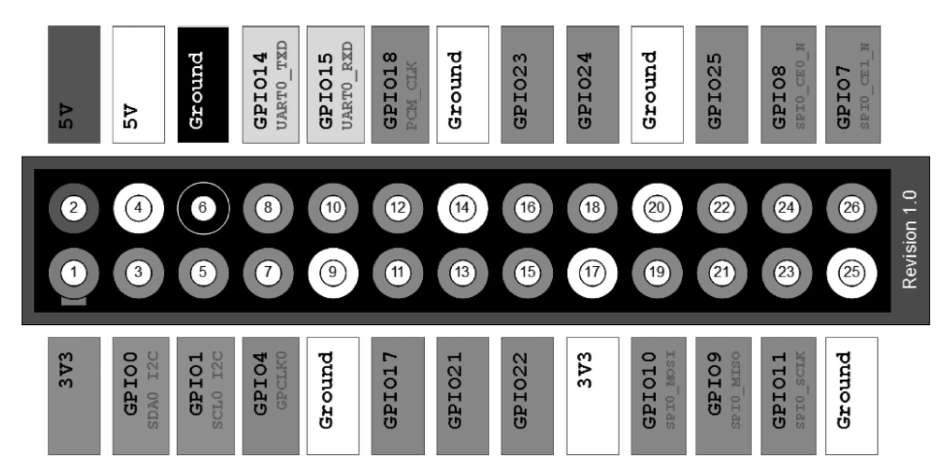


图 22-12 GPIO 引脚号码对应图

```
pi@raspberrypi ~ $ sudo wget http://goo.gl/1BOfJ -O /usr/bin/rpi-update
--2014-09-15 20:56:57--  http://goo.gl/1BOfJ
Resolving goo.gl (goo.gl)... 74.125.224.142, 74.125.224.128, 74.125.224.129, ...
Connecting to goo.gl (goo.gl)|74.125.224.142|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://raw.github.com/Hexxeh/rpi-update/master/rpi-update [following]
--2014-09-15 20:56:57--  https://raw.github.com/Hexxeh/rpi-update/master/rpi-update
```

图 22-13 下载固件

STEP3：更新固件

更新固件，完成后，重启树莓派机器，如图 22-14 所示。

```
$ sudo rpi-update
$ sudo reboot
```

```
pi@raspberrypi ~ $ sudo rpi-update
 *** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
 *** Performing self-update
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7100  100  7100    0     0  28962      0 --:--:-- --:--:-- --:--:-- 43030
 *** Relaunching after update
 *** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
 *** We're running for the first time
 *** Backing up files (this will take a few minutes)
 *** Backing up firmware
 *** Backing up modules 3.12.26+
```

图 22-14 更新固件

STEP4：创建工作路径和下载 TFT Kernel

输入以下命令，创建工作路径。

```
$ mkdir tft
$ cd tft/
```

下载 TFT LCD 的 Kernel，如图 22-15 所示。

```
$ wget http://ozzmaker.com/download/1.8TFT_Kernel_3.6.11.tar
```

```
pi@raspberrypi ~ $ mkdir tft
pi@raspberrypi ~ $ cd tft
pi@raspberrypi ~/tft $ wget http://ozzmaker.com/download/1.8TFT_Kernel_3.6.11.tar
--2014-09-15 21:00:29--  http://ozzmaker.com/download/1.8TFT_Kernel_3.6.11.tar
Resolving ozzmaker.com (ozzmaker.com)... 166.62.5.38
Connecting to ozzmaker.com (ozzmaker.com)|166.62.5.38|:80... connected.
```

图 22-15 创建工作路径和下载 TFT Kernel

STEP5：解压缩 TFT Kernel

输入以下命令，进行解压缩。

```
$ tar xvf 1.8TFT_Kernel_3.6.11.tar
```

STEP6：复制 TFT Kernel

输入以下命令，复制文件到对的系统路径，并删除下载的文件，如图 22-16 所示。

```
$ rm 1.8TFT_Kernel_3.6.11.tar
$ sudo cp -r * /
$ sudo reboot
```

```
pi@raspberrypi ~/tft $ rm 1.8TFT_Kernel_3.6.11.tar
pi@raspberrypi ~/tft $ ls
boot  lib
pi@raspberrypi ~/tft $ sudo cp -r * /
pi@raspberrypi ~/tft $ 
```

图 22-16 解压缩 TFT Kernel

5. 结果

完成后,重新开机就可以看到 TFT LCD 上在执行完整的 Linux 命令,如图 22-17 所示。

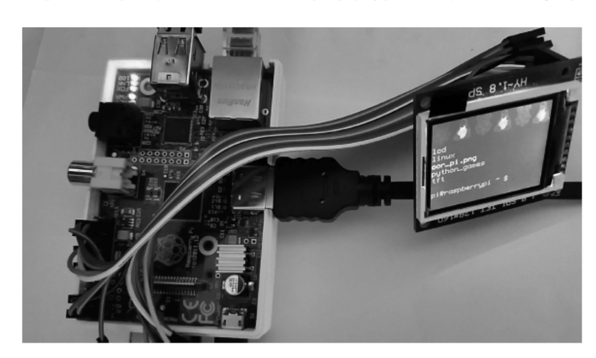


图 22-17　TFT LCD 执行结果

但是修改过后,会把 HDMI 的输出变得怪怪的,如图 22-18 所示。

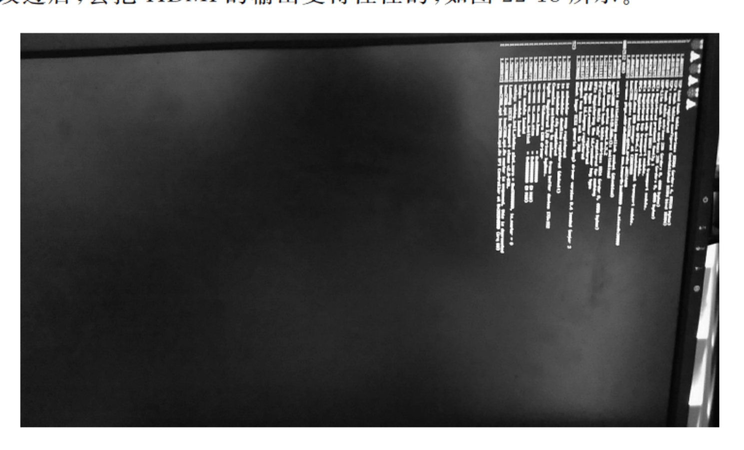


图 22-18　HDMI 的画面会保持这样

6. 教学影片

完整的教学视频可以看 RaspberryPi_TFTLCD3。

注意

输入以下命令在 RaspberryPi_TFTLCD3 中播放影片 file.h264,如图 22-19 所示。

```
$ sudo mplayer -vo fbdev2:/dev/fb1 -x 128 -y 160 -zoom file.h264
```

输入以下命令在 RaspberryPi_TFTLCD3 显示照片 file.jpg,如图 22-20 和图 22-21 所示。

```
$ sudo apt-get install fbi
$ sudo fbi -noverbose -T 1 -a -d /dev/fb1 file.jpg
```

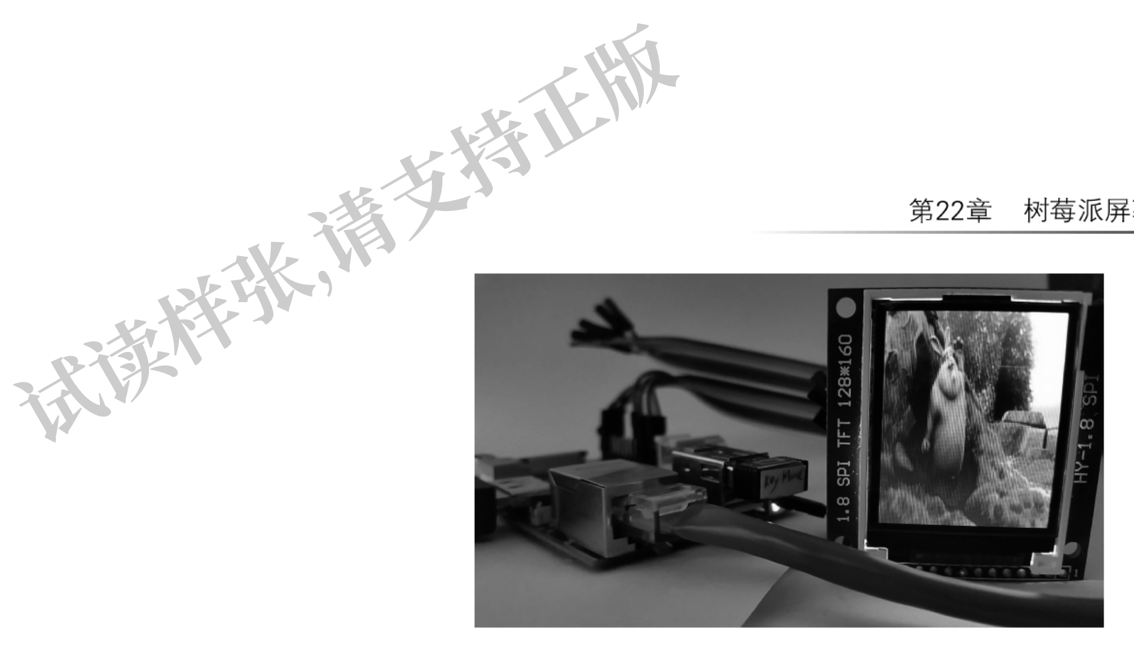


图 22-19　显示影片

```
pi@raspberrypi ~ $ sudo apt-get install fbi
Reading package lists... Done
```

图 22-20　安装 fbi

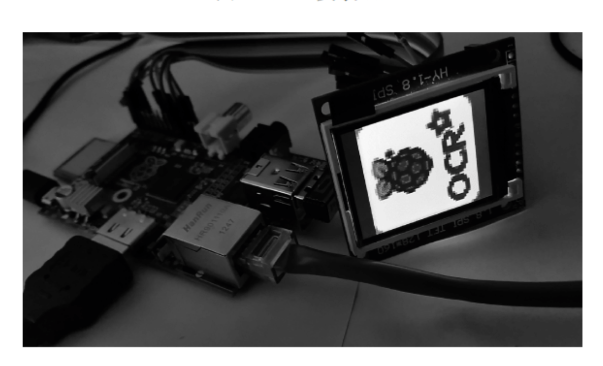


图 22-21　显示图片

22.3　在 TFT LCD 显示图文

案例 114

1. 介绍

本案例目的是通过 Python 命令，在 TFT LCD 上画出图片、文字、线条等。这样方面以后的应用，如果需要显示简单的图文数据，就可以按照下面的样例程序进行调整修改。

2. 设备

需要有以下硬件：

（1）树莓派开发板；

（2）1 个 TFT LCD；

(3) 面包板;

(4) 接线若干。

3. 接线

同案例。

4. 步骤

首先完成“22.2 使用 SPI 的 LCD TFT”。

STEP1:更新安装软件

输入以下命令,更新安装软件。

```
$ sudo apt - get update
```

STEP2:下载对应 Python 模块

输入以下命令安装 Python 相关的模块,如图 22-22~图 22-24 所示。

```
$ sudo apt - get install python - dev python - pip
$ sudo pip install wiringpi2
$ sudo apt - get install python - imaging
```

```
pi@raspberrypi ~/code $ sudo apt-get install python-dev python-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-dev is already the newest version.
python-dev set to manually installed.
```

图 22-22 安装 Python 相关的 python-dev 模块

```
pi@raspberrypi ~/code $ sudo pip install wiringpi2
Downloading/unpacking wiringpi2
  Downloading wiringpi2-1.0.10.tar.gz (72Kb): 72Kb downloaded
  Running setup.py egg_info for package wiringpi2
```

图 22-23 安装 Python 相关的 wiringpi2 模块

```
pi@raspberrypi ~/code $ sudo apt-get install python-imaging
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

图 22-24 安装 Python 相关的 python-imaging 模块

STEP3:编写代码

编写 Python 程序,在 TFT LCD 上画出文字和线条。

```
$ sudo nano TFT_LCD.py
```

样例程序:sample\ch21\TFT_LCD.py

```
1.  #!/usr/bin/env python
2.  # author: Powen Ko
```

```
3.  import wiringpi2 as wiringpi
4.  from subprocess import call
5.  import pygame,sys,os,time
6.  from pygame.locals import *
7.  os.environ["SDL_FBDEV"] = "/dev/fb1"
8.
9.  pygame.init()
10.
11. # 设定视窗
12. DISPLAYSURF = pygame.display.set_mode((128,160),0,32)
13. pygame.mouse.set_visible(0)
14. pygame.display.set_caption('Drawing')
15.
16. # 设定颜色
17. BLACK = (  0,   0,   0)
18. WHITE = (255,255,255)
19. RED   = (255,   0,   0)
20. GREEN = (  0,255,   0)
21. BLUE  = (  0,   0,255)
22.
23. # 画方块
24. DISPLAYSURF.fill(BLACK)
25. print "black"
26. time.sleep(.5)
27. pygame.display.update()
28.
29. DISPLAYSURF.fill(RED)
30. print "red"
31. time.sleep(.5)
32. pygame.display.update()
33.
34. DISPLAYSURF.fill(GREEN)
35. print "green"
36. time.sleep(.5)
37. pygame.display.update()
38.
39. DISPLAYSURF.fill(BLUE)
40. print "blue"
41. time.sleep(.5)
42. pygame.display.update()
43.
44. DISPLAYSURF.fill(WHITE)
45. print "white"
46. time.sleep(.5)
47. pygame.display.update()
48.
49. # demo draw some shapes
50. pygame.draw.polygon(DISPLAYSURF,GREEN,((16,0),(111,106),(36,277),(56,27),(0,106)))
51. pygame.draw.line(DISPLAYSURF,BLUE,(60,60),(120,60),4)
52. pygame.draw.line(DISPLAYSURF,BLUE,(120,60),(60,120))
53. pygame.draw.line(DISPLAYSURF,BLUE,(60,120),(120,120),4)
```

```
54. pygame.draw.circle(DISPLAYSURF,BLUE,(40,50),20,0)
55. pygame.draw.ellipse(DISPLAYSURF,RED,(110,200,40,80),1)
56. pygame.draw.rect(DISPLAYSURF,RED,(100,150,100,50))
57. pygame.display.update()
58. time.sleep(.5)
59.
60. font = pygame.font.Font(None,36)
61. text = font.render("powenko",1,BLUE)
62. textpos = text.get_rect(centerx = DISPLAYSURF.get_width()/2)
63. DISPLAYSURF.blit(text,textpos)
64. pygame.display.update()
65. time.sleep(.5)
```

程序解说

第 7～14 行：初始化。

第 23～27 行：涂满全黑的颜色。

第 39～42 行：涂满全蓝的颜色。

第 44～47 行：涂满全白的颜色。

第 49～58 行：画圆、画线。

第 60～65 行：显示文字。

完成后，按下 Ctrl＋O 组合键存储，Ctrl＋X 组合键离开文本编辑器 nano。

STEP4：执行和测试

确认 TFT LCD 接到树莓派上，并执行以下命令。

```
$ sudo python TFT_LCD.py
```

5. 结果

在执行时，会出现如图 22-25 所示的图片和文字在 TFT LCD 上。

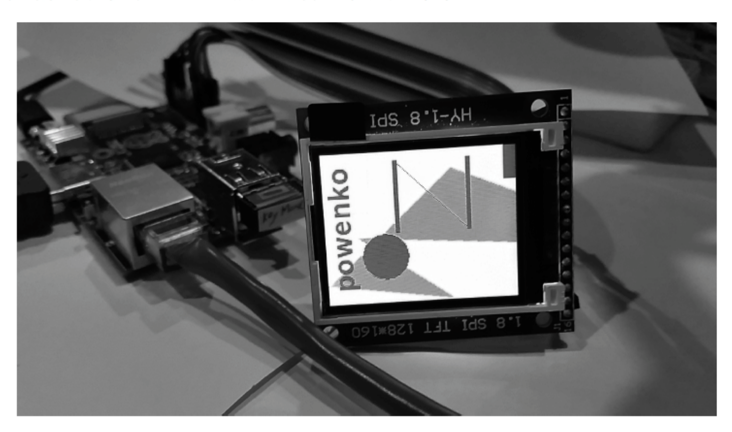


图 22-25　执行效果

6. 教学影片

完整的教学视频可以看 RaspberryPi_DigitalFrame。

22.4　数字相框

案例 115

1. 介绍

本案例目的是通过 Python 命令，在 TFT LCD 上显示照片和图片，必且能够自动搜寻指定目录下的图片。

2. 设备和接线

同案例 114。

3. 步骤

首先完成案例 114。

STEP1：编写代码

编写 Python 程序，在 TFT LCD 上显示照片和图片。

```
$ sudo nano DigitalFrame.py
```

样例程序：sample\ch21\DigitalFrame.py

```
1.   #!/usr/bin/env python
2.   # author: Powen Ko
3.
4.   import wiringpi2 as wiringpi
5.   from subprocess import call
6.   import pygame,sys,os,time
7.   from pygame.locals import *
8.
9.   os.environ["SDL_FBDEV"] = "/dev/fb1"
10.  w = 128
11.  h = 160
12.  pygame.init()
13.  # set up the window
14.  DISPLAYSURF = pygame.display.set_mode((w,h),0,32)
15.  pygame.mouse.set_visible(0)
16.  pygame.display.set_caption('Drawing')
17.
18.
19.  count = 0
20.  import os
21.  for root,dirs,files in os.walk("/"):
22.      for file in files:
```

```
23.         if (file.endswith(".png")) or (file.endswith(".jpg")) or (file.endswith(".gif")) :
24.             count = count + 1
25.             if count > 10 :
26.                 exit()
27.             print os.path.join(root,file)
28.             DISPLAYSURF.fill((0,0,0))
29.             img = pygame.image.load(os.path.join(root,file))
30.             img = pygame.transform.scale(img,(w,h))
31.             DISPLAYSURF.blit(img,(0,0))
32.             pygame.display.update()
33.             time.sleep(1)
34.
```

程序解说

第 9～16 行：初始化。

第 21～23 行：搜寻根目录下的所有图片。

第 29～32 行：显示图片在 TFT LCD 上。

完成后,按下 Ctrl+O 组合键存储,Ctrl+X 组合键离开文本编辑器 nano。

STEP2：执行和测试

确认 TFT LCD 接到树莓派上,并执行以下命令。

```
$ sudo python DigitalFrame.py
```

4. 结果

在执行时,程序会自动搜索根目录下的所有图片,并显示 10 张,每张以 0.5 秒的时间显示在 TFT LCD 上,如图 22-26 所示。

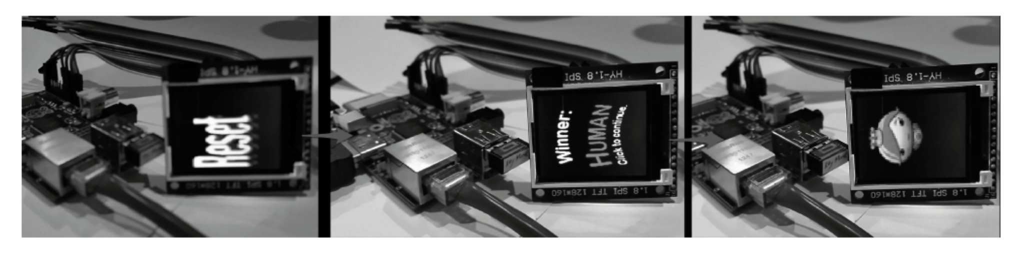


图 22-26　执行效果

5. 教学影片

完整的教学视频可以看 RaspberryPi_Draw。

第 23 章 树莓派超级计算机与计算机集群

23.1 树莓派 Cluster 结构

案例 116

树莓派最厉害的地方是作为一个小计算机具有超便宜的价格,但很多人都希望它的速度更快,甚至成为超级计算机,但是超级计算机的价格都非常高,那怎么把树莓派变成一台超级计算机呢?

什么是 Cluster 呢?就是有很多台计算机通过网络线连接在一起,但也可以分布在不同的地方。最重要是能让用户感觉到它就是一台超快速的计算机,并且通过其他计算机来合作与分布计算来达成目的。以现实举例就像各位到 Google 网站上搜索数据,根本感觉不到搜索的结果,来自成千上万台计算机中计算出的结果。

现在企业使用的 Cluster 结构有两种:

一种是所谓的平行运算,大家熟知的气象预报,当计算机中有各式各样的气候数据,按照未来的可行性使用大型计算机来计算预计结果,但是现在的数据越来越多,于是就把这些大量运算的工作放到 Cluster 里面,使用其他计算机的 CPU 来进行同步运算。

另一种是网页结构的 Cluster System,刚刚提到的搜索引擎,这种结构把各种各样的数据放在不同的主机和数据库上,有多个机器处理同一个项目,并且把数据分布备份到不同的机器上(也就是云数据库)。

还有一种是文件的部分,把文件分布备份,一般来说是三份,当其中一台存储空间有问题时,便由另外两台把数据整合撤销。

因为考虑稳定性和系统的维护和开放性,当前通常是用在 UNIX 系统上,当然最好使用同一个硬件结构和同样的软件,这样在维护上会最方便,现在树莓派上有很多选择可以搭建 Cluster 软件。以下假设采用平行运算的 Cluster,M(Master 主人/经理)当收到客户的要求时,会把工作通过网络分布成不同的小应用给不同的 S(slave 下属/员工),当 S 工作完成后,也一样把结果回传给 M,统一汇整之后再交给客户,如图 23-1 所示。

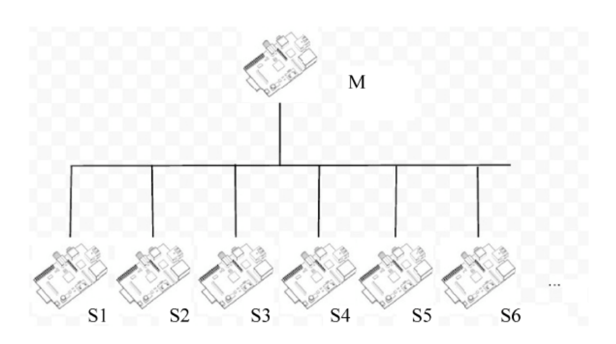


图 23-1 Cluster System 结构图

例如 Boise State University 爱达荷州的博伊西大学就用树莓派搭建了一组 RpiCluster 的计算机,如图 23-2 所示。

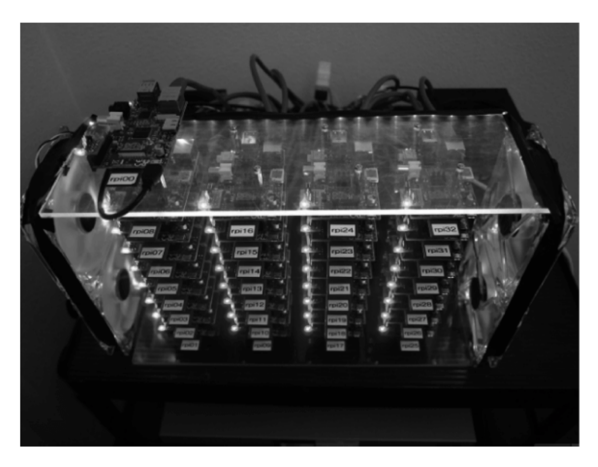


图 23-2 RpiCluster 计算机 Beowulf

23.2 树莓派 Riak Cluster 分布式数据库

23.2.1 树莓派 Riak Cluster 分布式数据库系统搭建

案例 117

1. 目的

本案例将会通过数个树莓派来搭建 Riak Cluster,如果手上没有两个以上的机器也没关系,就是加上节点 node 的部分不要执行,也能正常地运作。如果经费允许,还是推荐两台以上机器可以看出效果。本章先设置一台树莓派 Cluster 可以正确地工作。

Riak 是一个开放式代码的分布式数据库,Riak 的结构为:

(1) 可用性:Riak 复制和检索数据,因此可用于读入和写入操作,即使在有故障情况。

(2) 容错:不用担心硬盘坏掉或数据遗失的问题。

(3) 操作简单:通过 curl 的指定操作。

(4) 可扩展性:Riak 自动分布周围数据,并增加执行的效率。

2. 设备

(1) 树莓派开发板;

(2) 网络设备。

3. 步骤

STEP1:准备 Raspbian 的 SD 卡

官方推荐使用 Raspbain 的操作系统,所以按照树莓派数量准备数张 Raspbian 的 SD 卡,用来打开树莓派,并且完成网络和键盘等树莓派的一般开机的设置,本案例先设置其中的树莓派,确认可以正确的工作,在下一案例再介绍怎样把多台树莓派结合在一起工作。

STEP2:修改 /etc/apt/sources.list 设置

输入以下命令,打开文本编辑器 nano。

```
$ sudo nano /etc/apt/sources.list
```

把原本的删除,并且修改如下,如图 23-3 所示。

```
deb http://binaries.erlang-solutions.com/debian wheezy contrib
```

```
  GNU nano 2.2.6              File: /etc/apt/sources.list                    Modified

# deb http://mirrordirector.raspbian.org/raspbian/ wheezy main contrib non-free rpi
deb http://binaries.erlang-solutions.com/debian wheezy contrib


^G Get Help   ^O WriteOut   ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

图 23-3 调整后的/etc/apt/sources.list

STEP3:添加 Erlang Solutions 的公开钥匙

添加 Erlang Solutions 的公开钥匙到 apt-key 钥匙中,如图 23-4 所示。

```
$ cd ~
$ wget -O - http://binaries.erlang-solutions.com/debian/erlang_solutions.asc | sudo apt-key add -
```

```
pi@raspberrypi ~ $ bash
pi@raspberrypi ~ $ wget -O - http://binaries.erlang-solutions.com/debian/erlang_solutions.asc | su
do apt-key add -
```

图 23-4 下载 erlang_solutions

STEP4：安装 esl-erlang 软件

安装 esl-erlang 软件，因为需要安装 Erlang Solutions 的软件平行导向的处理，如图 23-5 所示，Erlang 是一种通用的平行导向程序语言，由乔·阿姆斯特朗(Joe Armstrong)在瑞典电信设备制造商爱立信的计算机科学研究室开发，目的是创造一种可以应付大规模开发的程序语言和执行环境。1987 年，Erlang 推出正式版本，最早是爱立信公司的内部软件，但经过十年的发展，于 1998 年发布开放源码版本。

```
pi@raspberrypi ~ $ sudo apt-get install esl-erlang
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

图 23-5 安装 esl-erlang 软件

在安装完毕后，重新引导树莓派。

```
$ sudo apt-get update
$ sudo apt-get install esl-erlang
$ sudo reboot
```

STEP5：下载 riak 的源程序

下载 riak 的源程序，如图 23-6 所示。

```
$ curl -O http://s3.amazonaws.com/downloads.basho.com/riak/1.2/1.2.1/riak-1.2.1.tar.gz
$ tar zxvf riak-1.2.1.tar.gz
$ cd riak-1.2.1
```

```
pi@raspberrypi ~ $ curl -O http://downloads.basho.com.s3-website-us-east-1.amazonaws.com/riak/CURRENT/ria
k-2.0.0.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 15.7M  100 15.7M    0     0  2328k      0  0:00:06  0:00:06 --:--:-- 3581k
pi@raspberrypi ~ $ tar zxvf riak-2.0.0.tar.gz
```

图 23-6 下载 riak 的源程序

注意

如果 curl 不能顺利下载文件，推荐用浏览器看一下现在最新的版本编号与位置。网址为 http://downloads.basho.com.s3-website-us-east-1.amazonaws.com/riak/CURRENT/。或

者可以试试自行调整因版本编号的变动,所产生的路径名称不同,如图 23-7 所示。



图 23-7　下载 riak 的源程序网站

STEP6:安装 git 版本管理软件

从版本管理服务器 Git 下载源程序,输入以下命令安装软件,如图 23-8 所示。

```
$ sudo apt-get install git
```

```
pi@raspberrypi ~/riak-1.2.1 $ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version.
git set to manually installed.
```

图 23-8　安装 git 版本管理软件

STEP7:编译程序 riak 的源程序

将 riak 源程序进行编译,输入以下命令完成,如图 23-9 所示。

```
$ cd rel
$ ./../rebar generate
```

```
pi@raspberrypi ~/riak-1.2.1/rel $ ./../rebar generate
==> rel (generate)
```

图 23-9　执行分配文件

输入以下命令,创建编译环境,如图 23-10 所示。

```
$ cd ..
$ make rel
```

STEP8:引导 riak

完成后,输入以下命令引导 riak。

```
$ ./rel/riak/bin/riak start
```

```
pi@raspberrypi ~/riak-1.2.1 $ make rel
fatal: Not a git repository (or any of the parent directories): .git
./rebar get-deps
==> cluster_info (get-deps)
==> lager (get-deps)
==> poolboy (get-deps)
==> meck (get-deps)
==> protobuffs (get-deps)
```

图 23-10 编译程序 riak 的源程序

4. 测试

先测试中文的树莓派是否可以正确工作,输入以下命令测试是否顺利引导 riak,如图 23-11 所示。

```
$ curl http://localhost:8098/ping
$ curl -XPUT http://localhost:8098/riak/hello/fr -d 'Allo'
$ curl http://localhost:8098/riak/hello/fr
```

这里的 8098 是 riak 的网络端口号。

```
pi@raspberrypi ~/riak-1.2.1 $ ./rel/riak/bin/riak start
pi@raspberrypi ~/riak-1.2.1 $  curl http://localhost:8098/ping
OKpi@raspberrypi ~/riak-1.2.1 $  curl -XPUT http://localhost:8098/riak/hello/fr -d 'Allo'
pi@raspberrypi ~/riak-1.2.1 $ curl http://localhost:8098/riak/hello/fr
'Allo'pi@raspberrypi ~/riak-1.2.1 $ █
```

图 23-11 执行结果

或者输入以下命令做测试。

```
$ curl http://localhost:8098/riak/druplets?keys=true | json_pp
```

表 23-1 所示为执行后的测试结果。

表 23-1 执行后的测试结果

全部	收到	录/传输标头文件	平均下载	上传速度	总额时间	花费时间	剩余时间	现在速度
100%(436)	100%(436)	0/0	564	0	--:--:--	--:--:--	--:--:--	599

5. 教学影片

完整的教学视频可以看 RaspberryPi_Riak Cluster-1 和 RaspberryPi_Riak Cluster-2。

注意

为了下次开机可以自动执行,先输入以下命令,增加这个程序的执行权限。

```
sudo chmod +x /home/pi/gpiokey/pikeyd-master/pikeyd
```

打开文本编辑器。

```
$ sudo nano /etc/rc.local
```

在最后# exit 0#之前加上以下的文字,用来设置执行 daemon 就是守护程序。

```
sudo /home/pi/ riak-1.2.1/rel/riak/bin/riak start &
```

23.2.2　多台树莓派一起处理分布式数据库

案例 118

1. 目的

本案例将会通过数个树莓派来搭建多台 Riak Cluster,如果没有两个以上的机器也没关系,一台也可以正常工作。

2. 步骤

首先完成“23.2.1 树莓派 Riak Cluster 分布式数据库系统搭建”。

STEP1:准备固定 IP 地址

因为 Cluster 只使用固定 IP 地址来彼此交换数据,所以每一台树莓派必须要有自己固定的 IP 地址,以方便数据的交换。详细的原理可以看本书“3-1 设置固定 IP 网络”的介绍。

输入以下命令,获取现在的 IP 网络地址、broadcast (Bcast)广播地址、netmask(Mask)网络地址掩码。

```
$ ifconfig
```

输入以下命令,获取 gateway(就是 default via 后的地址,以作者为例就是 192.168.0.1)

```
$ ip route show
```

请在文本模式下输入以下命令,就会进去 nano 文本编辑模式,编辑/etc/network/interfaces 文件。

```
$ sudo nano /etc/network/interfaces
```

输入和修改 eth0 的设置,就是第 5 行到第 9 行,address (固定 IP 地址),如果在家中,就可以设置 192.168.1.3 到 192.168.1.253 之间的地址,其他部分,按照步骤 1 得到的地址填入。如果该设备在公司或学校,请和网管人员沟通,他们会给你合适的地址。

```
1.
2.  auto lo
3.  iface lo inet loopback
4.
5.  iface eth0 inet static
6.  address 192.168.0.90
7.  gateway 192.168.0.1
8.  netmask 255.255.255.0
```

```
9.  broadcast 192.168.0.255
10.
11. allow-hotplug wlan0
12. …
```

完成后，按下键盘 Ctrl+O 组合键和 Enter 键存储文件，并按下 Ctrl+X 组合键离开 nano 文本编辑器。

STEP2：准备其他树莓派

这样就已经完成一台的设置，同样的步骤，按照“23.2.1 树莓派 Riak Cluster 分布式数据库系统搭建”和设置固定 IP 的方法，搭建其他树莓派。

注意

SD 卡的内容几乎是一样的，只是固定 IP 是不同的 IP 地址，但架设在同一个网域之中。

以作者为例，分别搭建的两台树莓派，IP 分别为：

192.168.0.90 和 92.168.0.91

STEP3：把 Cluster 开机

准备好，把所有的 Cluster 树莓派开机，如图 23-12 所示。

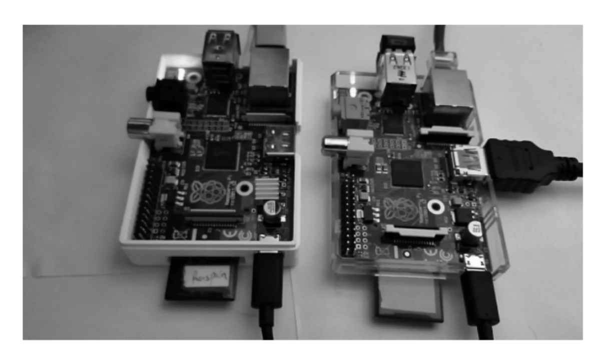


图 23-12 把 Cluster 开机

STEP4：设置 vm.args 的 IP 地址

对每一台树莓派，打开 rel/riak/etc/vm.args 设置 IP 地址，如图 23-13 所示。

```
$ ifconfig
$ cd rel/
$ nano riak/etc/vm.args
```

STEP5：设置 app.config 的 IP 地址

修改每一台的 riak/etc/app.config 配置文件。

```
$ nano riak/etc/app.config
```



```
GNU nano 2.2.6              File: etc/vm.args                          Modified

## Name of the riak node
-name riak@192.168.0.90

## Cookie for distributed erlang.  All nodes in the same cluster
## should use the same cookie or they will not be able to communicate.
-setcookie riak

## Heartbeat management; auto-restarts VM if it dies or becomes unresponsive
## (Disabled by default..use with caution!)
##-heart

## Enable kernel poll and a few async threads
+K true
+A 64

## Treat error_logger warnings as warnings
+W w

## Increase number of concurrent ports/sockets
-env ERL_MAX_PORTS 4096

## Tweak GC to run more often
-env ERL_FULLSWEEP_AFTER 0

^G Get Help   ^O WriteOut   ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

图 23-13　设置 IP 地址

修改其中的 127.0.0.1 为所有的树莓派的 IP 地址。记得每一台都要修改,并且重新开机,如图 23-14 所示。

```
GNU nano 2.2.6              File: etc/app.config                       Modified

%% -*- mode: erlang;erlang-indent-level: 4;indent-tabs-mode: nil -*-
%% ex: ft=erlang ts=4 sw=4 et
[
 %% Riak Client APIs config
 {riak_api, [
            %% pb_backlog is the maximum length to which the queue of pending
            %% connections may grow. If set, it must be an integer >= 0.
            %% By default the value is 5. If you anticipate a huge number of
            %% connections being initialised *simultaneously*, set this number
            %% higher.
            %% {pb_backlog, 64},

            %% pb_ip is the IP address that the Riak Protocol Buffers interface
            %% will bind to.  If this is undefined, the interface will not run.
            {pb_ip,   "192.168.0.90/91" },

            %% pb_port is the TCP port that the Riak Protocol Buffers interface
            %% will bind to
            {pb_port, 8087 }
            ]},

 %% Riak Core config
 {riak_core, [
              %% Default location of ringstate

^G Get Help   ^O WriteOut   ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

图 23-14　设置 IP 地址

STEP6:引导 riak

完成后,输入以下命令,先删除数据,再引导 riak,如图 23-15 所示。

```
$ cd /riak-1.2.1/rel/riak
$ rm -rf data
$ ./bin/riak ping
$ ./bin/riak start
```

```
pi@raspberrypi ~/riak-1.2.1/rel/riak $ ./bin/riak ping
Node 'riak@192.168.0.90' not responding to pings.
pi@raspberrypi ~/riak-1.2.1/rel/riak $ ./bin/riak start
```

图 23-15　引导 riak

输入以下命令进行测试,请把 IP 修改为实际的 IP 地址,并测试是否传回,如图 23-16 所示。

```
$ curl http://192.168.0.91:8098/ping
```

```
pi@raspberrypi ~/riak-1.2.1/rel/riak $ curl http://192.168.0.91:8098/ping
OKpi@raspberrypi ~/riak-1.2.1/rel/riak $
```

图 23-16　命令测试

STEP7:主机的设置

先决定哪一台要当主机,以作者为例,计划把 192.168.0.90 当(Master 主人/经理),把 192.168.0.91 当 S(slave 下属/员工),如图 23-17 所示。

```
$ ./bin/riak-admin cluster join riak@192.168.0.91
```

```
pi@raspberrypi ~/riak-1.2.1/rel/riak $ ./bin/riak-admin cluster join riak@192.168.0.91
Success: staged join request for 'riak@192.168.0.90' to 'riak@192.168.0.91'
```

图 23-17　加上 node 的设置

3. 测试

输入以下命令,确认现在连接的设备有哪些,如图 23-18 所示。

```
$ ./bin/riak-admin member-status
```

```
pi@raspberrypi ~/riak-1.2.1/rel/riak $ ./bin/riak-admin member-status
================================= Membership ==================================
Status     Ring    Pending    Node
-------------------------------------------------------------------------------
joining     0.0%      --      'riak@192.168.0.90'
valid     100.0%      --      'riak@192.168.0.91'
-------------------------------------------------------------------------------
Valid:1 / Leaving:0 / Exiting:0 / Joining:1 / Down:0
```

图 23-18　连接设备的列表

先测试树莓派是否可以正确工作,输入以下命令测试是否顺利执行 riak,并把 IP 地址转换成实际的主机地址,如图 23-19 所示。

```
$ curl http://192.168.0.90:8098/ping
$ curl -XPUT http://192.168.0.90:8098/riak/hello/fr -d 'Allo'
$ curl http://192.168.0.90:8098/riak/hello/fr
```

这里的 8098 是 riak 的网络端口号。

```
pi@raspberrypi ~/riak-1.2.1/rel/riak $ curl http://192.168.0.90:8098/ping
OKpi@raspberrypi ~/riak-1.2.1/rel/riak curl -XPUT http://192.168.0.90:8098/riak/hello/fr -d 'Allo'
pi@raspberrypi ~/riak-1.2.1/rel/riak $ curl http://192.168.0.90:8098/riak/hello/fr
'Allo'pi@raspberrypi ~/riak-1.2.1/rel/riak $
pi@raspberrypi ~/riak-1.2.1/rel/riak $
```

图 23-19 执行结果

或者输入以下命令做测试。

```
$ curl http://192.168.0.90:8098/riak/druplets?keys = true | json_pp
```

结果会看到执行速度从原本的 599 增加到 2096,下载的速度由原本的 564 到现在的 1800,如图 23-20 和表 23-2 执行后的测试结果。

```
pi@raspberrypi ~/riak-1.2.1/rel/riak $ curl http://192.168.0.90:8098/riak/druplets?keys=true | json_pp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   436  100   436    0     0   1800      0 --:--:-- --:--:-- --:--:--  2096
{
   "keys" : [],
   "props" : {
      "w" : "quorum",
      "r" : "quorum",
      "young_vclock" : 20,
      "big_vclock" : 50,
      "old_vclock" : 86400,
      "n_val" : 3,
      "postcommit" : [],
      "small_vclock" : 50,
      "last_write_wins" : false,
      "allow_mult" : false,
      "rw" : "quorum",
      "pw" : 0,
      "dw" : "quorum",
      "name" : "druplets",
```

图 23-20 执行结果

表 23-2 执行后的测试结果

全部	收到	录/传输标头文件	平均下载	上传速度	总额时间	花费时间	剩余时间	现在速度
100 436	100 436	0 0	1800	0	--:--:--	--:--:--	--:--:--	2096

4. 教学影片

完整的教学视频可以看 RaspberryPi_Riak Cluster_AddNode-1 和 RaspberryPi_Riak Cluster_AddNode-2。

23.3 分布式存储服务/分布式文件系统 DFS——GlusterFS

23.3.1 搭建分布式存储服务/分布式文件系统 DFS——GlusterFS

案例 119

1. 分布式存储服务 DFS

以前存储数据的时候,可能要准备一个大硬盘,把要备份的数据存放在上面,或者使用

软件(例如苹果操作系统上最出名的 Time Machine 软件)，自动备份整个工作硬盘并做数据压缩，如果系统出问题，可以回到过去的时间点把相关的数据取回。大公司可以准备多个超大硬盘，让员工自行处理里面的文件数据。

现在有更好的方法——分布式文件系统(Distributed File System，DFS)可以让网管人员更轻松地备份数据，而不用担心硬盘损坏，造成数据的永久遗失。

对存储在网络上的文件来说，分布式文件系统或网络文件系统(Network File System)，是一种允许文件通过网络在多台主机上分享的文件系统，可让多机器上的多用户分享文件和存储空间。

但在这样的文件系统中，客户端并非直接访问底层的数据存储块，而是通过网络，以特定的通信协议和服务器沟通。借助通信协议的设计，可以让客户端和服务器端都能根据访问控制清单或是授权来限制对于文件系统的访问。相对地，在一个共享的磁盘文件系统中，所有节点对数据存储块都有相同的访问权。在这样的系统中，访问权限就必须由客户端程序来控制。分布式文件系统最大的优点就是，当其中一个硬盘出现问题时，还可通过备份的文件继续运作，系统不会有数据损失。

2. 目的

本案例将会用树莓派来假设鼎鼎大名的 GlusterFS 系统，如图 23-21 所示。



图 23-21 GlusterFS 网站

3. 步骤

STEP1：更新安装程序

自行下载用来更新树莓派的 apt-get 安装程序。

```
$ sudo apt - get update
$ sudo apt - get upgrade
```

STEP2：gluster 下载文件

到官方网站确认最新的下载网址，如图 23-22 所示。



www.gluster.org/download/

GLUSTER About Blog Documentation Community Forge

Downloads
Latest
Gluster 3.5
Gluster 3.4
Gluster Ubuntu PPA's
Gluster Core (forge)

图 23-22　到官方网站确认下载的网址

下载以下 3 个文件,如图 23-23 所示。

```
$ wget http://download.gluster.org/pub/gluster/glusterfs/3.5/3.5.0/Debian/apt/pool/main/
g/glusterfs/glusterfs_3.5.0.orig.tar.gz
$ wget http://download.gluster.org/pub/gluster/glusterfs/3.5/3.5.0/Debian/apt/pool/main/
g/glusterfs/glusterfs_3.5.0-1.dsc
$ wget http://download.gluster.org/pub/gluster/glusterfs/3.5/3.5.0/Debian/apt/pool/main/
g/glusterfs/glusterfs_3.5.0-1.debian.tar.gz
```

```
pi@raspberrypi ~/glusterfs $ wget http://download.gluster.org/pub/gluster/glusterfs/3.5/3.5.0/Debian/apt/
pool/main/g/glusterfs/glusterfs_3.5.0-1.debian.tar.gz
--2014-09-02 23:47:37--  http://download.gluster.org/pub/gluster/glusterfs/3.5/3.5.0/Debian/apt/pool/mair
/g/glusterfs/glusterfs_3.5.0-1.debian.tar.gz
Resolving download.gluster.org (download.gluster.org)... 50.57.69.89
Connecting to download.gluster.org (download.gluster.org)|50.57.69.89|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12215 (12K) [application/x-gzip]
Saving to: `glusterfs_3.5.0-1.debian.tar.gz'

100%[===============================================================>] 12,215      --.-K/s   in 0.001s

2014-09-02 23:47:38 (19.5 MB/s) - `glusterfs_3.5.0-1.debian.tar.gz' saved [12215/12215]
```

图 23-23　下载文件

STEP3：解压缩

切换到管理权限。

```
$ sudo su
```

进行解压缩。

```
# tar xzvf glusterfs_3.5.0.orig.tar.gz
# tar xzvf glusterfs_3.5.0-1.debian.tar.gz
```

通过 lse 确认,如图 23-24 所示。

```
root@raspberrypi:/home/pi/gluster# ls -al
total 4804
drwxr-xr-x  4 pi   pi      4096 Sep  4 04:55 .
drwxr-xr-x  5 pi   pi      4096 Sep  4 04:52 ..
drwxr-xr-x  4 root root    4096 Apr 24 03:24 debian
drwxr-xr-x 14  498  499    4096 Apr 17 07:18 glusterfs-3.5.0
-rw-r--r--  1 root root   12215 Apr 24 03:33 glusterfs_3.5.0-1.debian.tar.gz
-rw-r--r--  1 root root    1801 Apr 24 03:33 glusterfs_3.5.0-1.dsc
-rw-r--r--  1 root root 4883901 Apr 24 03:33 glusterfs_3.5.0.orig.tar.gz
```

图 23-24　解压缩

STEP4：迁移文件

迁移文件，如图 23-25 所示。

```
# mv debian glusterfs-3.5.0
# cd glusterfs-3.5.0/
```

```
root@raspberrypi:/home/pi/gluster# mv debian glusterfs-3.5.0
root@raspberrypi:/home/pi/gluster# cd glusterfs-3.5.0/
```

图 23-25　迁移文件

STEP5：安装所需要的软件

安装需要的软件，如图 23-26 所示。

```
# apt-get install devscripts
# apt-get install dh-autoreconf libfuse-dev libibverbs-dev libdb-dev attr flex bison
libreadline-dev
# apt-get install libncurses5-dev libssl-dev libxml2-dev python-all-dev liblvm2-dev
libaio-dev
# apt-get install librdmacm-dev chrpath hardening-wrapper
```

```
root@raspberrypi:/home/pi/gluster/glusterfs-3.5.0# apt-get install dh-autoreconf libfuse-dev libibverbs-de
v libdb-dev attr flex bison libreadline-dev libncurses5-dev libssl-dev libxml2-dev python-all-dev liblvm2-
dev libaio-dev librdmacm-dev chrpath hardening-wrapper
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

图 23-26　安装需要的软件

STEP6：编译程序

准备和编译程序，按照顺序执行以下 3 个命令，如图 23-27 所示，每个操作都会持续近半个小时，请耐心等待。

```
# debuild -us -uc
# make
# make install
```

```
root@raspberrypi:/home/pi/gluster/glusterfs-3.5.0# make install
Making install in argp-standalone
Making install in .
Making install in libglusterfs
Making install in src
```

图 23-27　编译程序

4. 结果

完成后,输入以下命令进行引导。

```
# /etc/init.d/glusterd start
```

并通过以下命令询问 gluster 现在情况。

```
# gluster peer status
```

会有回传值,显示下面的 node 节点还有多少台树莓派,因为这只有一台树莓派,所以显示为 0,如图 23-28 所示。

```
Number of Peers: 0
```

```
root@raspberrypi:/home/pi/gluster/glusterfs-3.5.0# /etc/init.d/glusterd start
[....] Starting glusterd service: glusterd
. ok
root@raspberrypi:/home/pi/gluster/glusterfs-3.5.0#
root@raspberrypi:/home/pi/gluster/glusterfs-3.5.0# gluster peer status
Number of Peers: 0
```

图 23-28　执行结果

注意

如果要设置开机自动引导,gluster 的操作可以通过以下命令设置。

```
# update-rc.d glusterd defaults
```

5. 教学影片

完整的教学视频可以看 RaspberryPi_gluster_01_build 和 RaspberryPi_gluster_02_test。

23.3.2　搭建分布式存储服务 DFS——GlusterFS

案例 120

1. 目的

本案例会通过数个树莓派来搭建多台 Gluster,如果手上没有两个以上的机器,也没关系,一台也可以正常工作。

2. 步骤

STEP1:前置作业

前置作业:

(1) 设置两台树莓派,都安装好 Gluster。

(2) 分别设置不同的 IP 固定地址。以作者为例,两台树莓派分别设置为 192.168.0.90 和 192.168.0.91。

(3) 重新开机,各自引导所有的树莓派和 gluster。

并且运行以下命令,如图 23-29 所示。

```
$ sudo /etc/init.d/glusterd start
```

```
pi@raspberrypi ~ $ sudo /etc/init.d/glusterd start
[ ok ] Starting glusterd service: glusterd.
```

图 23-29　引导 gluster

STEP2:添加其他的节点 node

在主要的机器上面,添加其他的节点 node,把以下的 IP 换成实际的地址,如图 23-30 所示。

```
$ sudo gluster peer probe 192.168.0.91
```

```
pi@raspberrypi ~ $ sudo gluster peer probe 192.168.0.91
peer probe: success.
```

图 23-30　添加其他的节点 node IP 地址

STEP3:确认节点情况

添加后,可以在主要的机器上查阅节点情况,显示当前丛集成员状态,如图 23-31 所示。

```
$ sudo su
# gluster peer status
```

```
root@raspberrypi:/# gluster peer status
Number of Peers: 1

Hostname: 192.168.0.91
Uuid: b8493098-7a77-4e36-a26a-b8820100b353
State: Peer in Cluster (Connected)
```

图 23-31　确认节点

STEP4:准备数据路径

参考本书"5.1 添加 USB 外接硬盘",在各个树莓派接上外接硬盘。

假设要用根目录的 storage1 和 storage2 来设置对应的目录,输入以下命令,分别在 192.168.0.90 执行

```
# mkdir storage1
```

和 192.168.0.91 执行

```
# mkdir storage2
```

STEP5：准备数据路径

在主机上添加对应的路径，命名路径为 test-volume，如图 23-32 所示。

```
# gluster volume create test - volume 192.168.0.90:/storage1 192.168.0.91:/storage2 force
```

```
root@raspberrypi:/# sudo gluster volume create test-volume 192.168.0.90:/storage1 192.168.
0.91:/storage2 force
volume create: test-volume: success: please start the volume to access data
root@raspberrypi:/# gluster peer status
```

图 23-32　添加路径

如果参考本书"5.1 添加 USB 外接硬盘"，mount 的路径为"/media/usbhdd"，就可以改为以下命令，把虚拟路径 test-volume 换个名称。

```
# gluster volume create test - volume 192.168.0.90:/media/usbhdd 192.168.0.91:/media/usbhdd
force
```

注意

可以自动对应路径，使用 gluster volume create，指的是 root 根目录的相对路径名称。

如果还想对应多台，可以在后面添加，例如

```
gluster volume create test - volume 192.168.0.90:/storage1 192.168.0.91:/storage2 192.168.0.
92:/storage3 192.168.0.93:/storage4 force
```

STEP6：确认

使用以下命令确定是否有加进来，如图 23-33 所示。

```
# gluster volume info
```

```
root@raspberrypi:/#  gluster volume info

Volume Name: test-volume
Type: Distribute
Volume ID: 2a3ac57c-1b9b-490c-bee6-069015cc0264
Status: Created
Number of Bricks: 2
Transport-type: tcp
Bricks:
Brick1: 192.168.0.90:/storage1
Brick2: 192.168.0.91:/storage2
```

图 23-33　确定是否有加进来

STEP7：引导该 Volume

输入以下命令，引导该 Volume，如图 23-34 所示。

```
# gluster volume start test - volume
```

```
root@raspberrypi:/# gluster volume start test-volume
volume start: test-volume: success
```

图 23-34　引导该 Volume

STEP8：创建对应的路径

把 glusterfs 的 IP 地址，创建一个对应的挂载点，方便实际实用时，可以使用该路径来访问数据，如图 23-35 所示。

```
# mkdir /mnt/test - volume
# mount.glusterfs 192.168.0.90:/test - volume /mnt/test - volume
```

```
root@raspberrypi:/mnt# ls
test-volume
root@raspberrypi:/mnt# cd ..
root@raspberrypi:/# mount.glusterfs 192.168.0.90:/test-volume /mnt/test-volume
```

图 23-35　创建对应的路径

完成后，输入以下命令确认是否挂载上，如图 23-36 所示。

```
# df - h
```

```
root@raspberrypi:/mnt/test-volume# df -h
Filesystem                Size  Used Avail Use% Mounted on
rootfs                    7.3G  3.0G  4.0G  43% /
/dev/root                 7.3G  3.0G  4.0G  43% /
devtmpfs                  215M     0  215M   0% /dev
tmpfs                      44M  244K   44M   1% /run
tmpfs                     5.0M     0  5.0M   0% /run/lock
tmpfs                      88M     0   88M   0% /run/shm
/dev/mmcblk0p1             56M  9.6M   47M  18% /boot
192.168.0.90:/test-volume  15G  5.9G  8.0G  43% /mnt/test-volume
```

图 23-36　文件系统

3. 测试

因为 gluster 是自动引导的服务，当设置完毕后，下次开机它便会自动引导。只要把文件放入/mnt/test-volume，就会自动和即时地备份到其他的树莓派上，如图 23-37 所示。

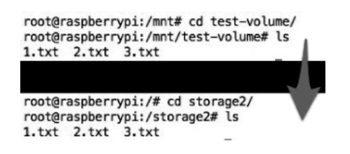


图 23-37　自动备份文件

4. 教学影片

完整的教学视频可以看 RaspberryPi_Gluster-Volume。

注意

1）技术扩展

作者自己在搭建时，把对应的路径指到 USB 的硬盘之外，还会挂载路径到云服务器，这样方便大家到云服务器访问备份数据，另外搭建 Samba 指定到挂载路径也是好方法。很多公司都会用这样的方法，备份 3 份到不同的地方，让文件更加安全，更加严谨的公司会把这 3 份文件放在不同的地点，以避免天灾人祸产生的数据遗失。

2）补充数据

如果要删除某备份点,输入以下命令就可以删除树莓派的备份,如图 23-38 所示。

```
# gluster volume remove - brick test - volume 192.168.0.91:/storage2
```

```
root@raspberrypi:/home/pi# gluster volume remove-brick test-volume  192.168.0.91:/storage2

WARNING: running remove-brick commands without an explicit option is deprecated, and will
be removed in the next version of GlusterFS.
To forcibly remove a brick in the next version of GlusterFS, you will need to use "remove-
brick force".
Removing brick(s) can result in data loss. Do you want to Continue? (y/n) y
volume remove-brick commit force: success
```

图 23-38　删除某备份点

如果要新加入某备份点,输入以下命令就可以加入树莓派的备份。

```
# gluster volume add - brick test - volume 192.168.0.91:/storage2
```

23.4　分布式树莓派网站服务器

案例 121

1. 目的

当网站达到一定的规模,就会有很多用户,除了把网站服务器升级、网络上下传的速度加快之外,还有其他的方法吗？这里柯博文老师将介绍使用多个树莓派通过网络衔接在一起,并且做所谓的流量平衡,可以把网站流量的负担,平均分布到不同的机器上。如果手上没有多台机器,也没关系,一台也可以正常地工作,不过要出效果,建议使用 3 台以上的机器,1 台当成代理和平衡服务器,剩下的树莓派当成网站服务器,这样网站服务器越多,效果越好。

2. 步骤

STEP1：前置作业

前置作业

(1) 参考本书的“4.1 搭建 PHP 网站服务器”,先把全部的机器都安装 PHP 网页的 Apache2 软件。

(2) 设置固定 IP 地址。

STEP2：设置 Proxy

指定一台当成 Proxy 伺服器,如图 23-39 所示。

```
$ sudo su
# a2enmod proxy
# a2enmod proxy_http
# a2enmod proxy_balancer
# service apache2 restart
```

```
root@raspberrypi:/mnt/test-volume# sudo a2enmod proxy
Module proxy already enabled
root@raspberrypi:/mnt/test-volume# sudo a2enmod proxy_http
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
To activate the new configuration, you need to run:
  service apache2 restart
root@raspberrypi:/mnt/test-volume# sudo a2enmod proxy_balancer
Considering dependency proxy for proxy_balancer:
Module proxy already enabled
Enabling module proxy_balancer.
To activate the new configuration, you need to run:
  service apache2 restart
root@raspberrypi:/mnt/test-volume# service apache2 restart
[....] Restarting web server: apache2apache2: Could not reliably determine the s
erver's fully qualified domain name, using 127.0.1.1 for ServerName
 ... waiting apache2: Could not reliably determine the server's fully qualified
domain name, using 127.0.1.1 for ServerName
. ok
root@raspberrypi:/mnt/test-volume#
```

图 23-39　设置 Proxy

STEP3：测试速度

安装测试网络速度的软件，并用 IP 地址来测试一下只有一台的网页执行速度，如图 23-40 所示。

```
$ sudo apt-get install siege
$ siege -c200 -d1 -r1 -v http://192.168.0.90/index.html
```

```
done.
Transactions:                  200 hits
Availability:               100.00 %
Elapsed time:                 1.94 secs
Data transferred:             0.03 MB
Response time:                0.01 secs
Transaction rate:           103.09 trans/sec
Throughput:                   0.01 MB/sec
Concurrency:                  1.01
Successful transactions:       200
Failed transactions:             0
Longest transaction:          0.04
Shortest transaction:         0.00
```

图 23-40　一台的测试速度

STEP4：修改 Proxy 设置

将 Proxy 伺服器那台，使用文本编辑器，修改设置。

```
$ sudo nano /etc/apache2/sites-enabled/000-default
```

添加 Proxy 的设置，在 ErrorLog ${APACHE_LOG_DIR}/error.log 之前，添加以下的设置，如图 23-41 所示。

```
ProxyRequests Off
<Proxy balancer://rpicluster>
        BalancerMember http://192.168.0.91:80
        BalancerMember http://192.168.0.92:80
        AllowOverride None
        Order allow,deny
```

```
        allow from all
        ProxySet lbmethod = byrequests
</Proxy>
```

```
 GNU nano 2.2.6      File: /etc/apache2/sites-enabled/000-default

<VirtualHost *:80>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride all
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all

        </Directory>
        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride all
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>
        ProxyRequests Off
        <Proxy balancer://rpicluster>
                BalancerMember http://192.168.0.91:80
                BalancerMember http://192.168.0.92:80
                AllowOverride None
                Order allow,deny
                allow from all
                ProxySet lbmethod=byrequests
        </Proxy>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

                             [ Wrote 38 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 23-41　修改 Proxy 设置

按照实际的 IP 修改网络地址，如果有多台机器，只要重复 BalancerMember 的设置就可以了。

```
BalancerMember http://192.168.0.92:80
```

STEP5：重新引导

输入以下命令，重新引导服务器。

```
$ sudo /etc/init.d/apache2 restart
```

3. 结果

完成后，再通过网页速度测试软件，如图 23-42 所示，同一个网址例如 200 次的网页需求，测试 Proxy 伺服器的那台 IP 地址，就会看出通过多台树莓派网站服务器，效果就会提

升,并且连接越多台效果越好。

```
$ siege -c200 -d1 -r1 -v http://192.168.0.90/index.html
```

```
done.
Transactions:                    200 hits
Availability:                 100.00 %
Elapsed time:                   1.60 secs
Data transferred:               0.03 MB
Response time:                  0.24 secs
Transaction rate:             125.00 trans/sec
Throughput:                     0.02 MB/sec
Concurrency:                   29.69
Successful transactions:         200
Failed transactions:               0
Longest transaction:            0.39
Shortest transaction:           0.02
```

图 23-42　一台的测试速度

4. 教学影片

完整的教学视频可以看 RaspberryPi_Proxy-Cluster。

23.5　多个树莓派平行处理——MPICH 软件

23.5.1　安装 MPICH 软件

案例 122

1. 目的

很多人会觉得树莓派的速度不够快,执行复杂的应用程序会花很多时间,在本案例介绍个技巧,让树莓派绝对可以跟超级计算机一较高下。MPICH 软件如图 23-43 所示,它是一款免费软件,用来处理在平行计算中的数据传递,这样设置完毕后,通过把多台计算机协同工作,加快计算速度。

2. 步骤

STEP1:前置作业

前置作业

分别给树莓派设置不同的固定 IP 地址。以作者为例,两台树莓派分别设置为 192.168.0.91 和 192.168.0.92。

STEP2:更新安装软件

输入以下命令,更新安装软件。

```
$ sudo apt-get update
```

STEP3:下载 MPICH 软件

输入以下的命令,创建路径和下载 MPICH 软件,如图 23-44 所示。



图 23-43　MPICH 官方网站

```
$ mkdir mpich2
$ cd ~/mpich2
$ wget http://www.mpich.org/static/downloads/3.1/mpich-3.1.tar.gz
```

```
pi@raspberrypi ~ $ mkdir mpich2
pi@raspberrypi ~ $ cd ~/mpich2
pi@raspberrypi ~/mpich2 $ wget http://www.mpich.org/static/downloads/3.1/mpich-3
.1.tar.gz
--2014-09-17 16:15:25--  http://www.mpich.org/static/downloads/3.1/mpich-3.1.tar
.gz
Resolving www.mpich.org (www.mpich.org)... 140.221.6.71
Connecting to www.mpich.org (www.mpich.org)|140.221.6.71|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11088555 (11M) [application/x-gzip]
Saving to: `mpich-3.1.tar.gz'

100%[=======================================>] 11,088,555  3.02M/s   in 4.0s

2014-09-17 16:15:30 (2.63 MB/s) - `mpich-3.1.tar.gz' saved [11088555/11088555]
```

图 23-44　下载 MPICH 软件

STEP4：解压缩

输入以下命令，把刚刚下载的 MPICH 源程序解压缩。

```
$ tar xfz mpich-3.1.tar.gz
```

STEP5：创建 3 个工作路径

创建 3 个工作路径，如图 23-45 所示。

```
$ sudo mkdir /home/rpimpi/
$ sudo mkdir /home/rpimpi/mpi-install
$ sudo mkdir /home/pi/mpi-build
```

```
pi@raspberrypi ~/mpich2 $ sudo mkdir /home/rpimpi/
pi@raspberrypi ~/mpich2 $ sudo mkdir /home/rpimpi/mpi-install
pi@raspberrypi ~/mpich2 $ sudo mkdir /home/pi/mpi-build
```

图 23-45　创建 3 个工作路径

STEP6：安装 gfortran 软件

输入以下命令，安装 gfortran 软件，如图 23-46 所示。

```
$ cd /home/pi/mpi-build
$ sudo apt-get install gfortran
```

```
pi@raspberrypi ~/mpich2 $ cd /home/pi/mpi-build
pi@raspberrypi ~/mpi-build $ sudo apt-get install gfortran
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
```

图 23-46　安装 gfortran

STEP7：做编译程序前的设置

输入以下命令，做编译程序前的设置，如图 23-47 所示。

```
$ sudo /home/pi/mpich2/mpich-3.1/configure -prefix=/home/rpimpi/mpi-install
```

```
pi@raspberrypi ~/mpi-build $ sudo /home/pi/mpich2/mpich-3.1/configure -prefix=/h
ome/rpimpi/mpi-install
Configuring MPICH version 3.1 with  '-prefix=/home/rpimpi/mpi-install'
Running on system: Linux raspberrypi 3.6.11+ #1 PREEMPT Wed Nov 13 15:56:48 PST
2013 armv6l GNU/Linux
```

图 23-47　做编译程序前的设置

STEP8：编译程序

输入以下命令，编译程序，如图 23-48 所示。

```
$ sudo make
```

```
pi@raspberrypi ~/mpi-build $ sudo make
if test ! -h ./src/include/mpio.h ; then \
    rm -f ./src/include/mpio.h ; \
    ( cd ./src/include &&       \
      ln -s ../mpi/romio/include/mpio.h ) ; \
fi
make  all-recursive
```

图 23-48　编译程序

STEP9：安装程序

输入以下命令，安装和编译程序，如图 23-49 所示。

```
$ sudo make install
```

```
pi@raspberrypi ~/mpi-build $ sudo make install
make  install-recursive
make[1]: Entering directory '/home/pi/mpi-build'
Making install in /home/pi/mpi-build/src/mpl
```

图 23-49　安装和编译程序

STEP10：修改 Path 工作路径

打开文本编辑器，修改. bashrc 文件。

```
$ cd ~
$ nano .bashrc
```

在最后加上如下内容，如图 23-50 所示。

```
PATH=$PATH:/home/rpimpi/mpi-install/bin
```

完成后，按下 Ctrl+O 组合键存储，并按下 Ctrl+X 组合键离开文本编辑器 nano。

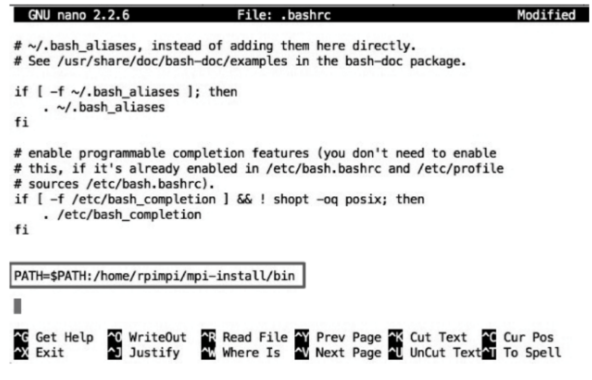


图 23-50　修改 Path 工作路径

3. 测试

重新开机后，输入以下命令 mpiexec，显示是否有正确反应，如图 23-51 所示，如果取到名字，就代表成功。

```
$ mpiexec -n 1 hostname
```

输入以下命令执行 MPICH 软件，如图 23-52 所示。

```
pi@raspberrypi ~ $ mpiexec -n 1 hostname
raspberrypi
```

图 23-51　通过 mpiexec 取得 hostname 的名称

```
$ cd ~/mpi-build/examples
$ ./cpi
```

```
pi@raspberrypi ~/mpi-build/examples $ cd ~/mpi-build/examples
pi@raspberrypi ~/mpi-build/examples $ ls
cpi  cpi.o  machinefile  Makefile
pi@raspberrypi ~/mpi-build/examples $ ./cpi
Process 0 of 1 is on raspberrypi
pi is approximately 3.1415926544231341, Error is 0.0000000008333410
wall clock time = 0.001324
```

图 23-52　执行 MPICH 软件

并通过以下命令编辑 machinefile 文件,如图 23-5 所示。

```
$ cd /home/rpimpi/mpi-install/bin
$ sudo nano machinefile
```

```
pi@raspberrypi ~ $  cd /home/rpimpi/mpi-install/bin
pi@raspberrypi /home/rpimpi/mpi-install/bin $ sudo nano machinefile
```

图 23-53　编辑 machinefile 文件

并添加机器本身的 IP 地址,如图 23-54 所示。

```
GNU nano 2.2.6            File: machinefile                    Modified

192.168.0.91

                              [ New File ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

图 23-54　机器本身的 IP 地址

mpiexec 目前可以执行 C 语言和 Fortran 语言,可以输入以下命令测试 mpiexec,如图 23-55 和图 23-56 所示。

```
$ cd /home/rpimpi/mpi-install/bin
$ mpiexec -f machinefile -n 1 ~/mpi-build/examples/cpi
$ mpiexec -f machinefile -n 2 ~/mpi-build/examples/cpi
```

```
pi@raspberrypi ~ $  cd /home/rpimpi/mpi-install/bin
pi@raspberrypi /home/rpimpi/mpi-install/bin $ mpiexec -f machinefile -n 1 ~/mpi-build/examples/cpi
Process 0 of 1 is on raspberrypi
pi is approximately 3.1415926544231341, Error is 0.0000000008333410
wall clock time = 0.001349
```

图 23-55　执行 C 语言的结果-1

```
pi@raspberrypi /home/rpimpi/mpi-install/bin $ mpiexec -f machinefile -n 2 ~/mpi-build/examples/cpi
Process 1 of 2 is on raspberrypi
Process 0 of 2 is on raspberrypi
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
wall clock time = 0.018689
```

图 23-56　执行 C 语言的结果-2

4. 教学影片

完整的教学视频可以看 RaspberryPi_mpich1 和 RaspberryPi_mpich2。

23.5.2　修改 hostname

案例 123

1. 目的

建议把其他连接上的树莓派，hostname 的名字由 raspberry 改成其他的名称，以方便管理。

2. 步骤

STEP1：修改 /etc/hosts

修改 hostname 需要通过文本编辑器，修改 /etc/hosts 的配置文件。

```
sudo nano /etc/hosts
```

把 127.0.1.1 后的 raspberry 改成其他的名称，如图 23-57 所示。

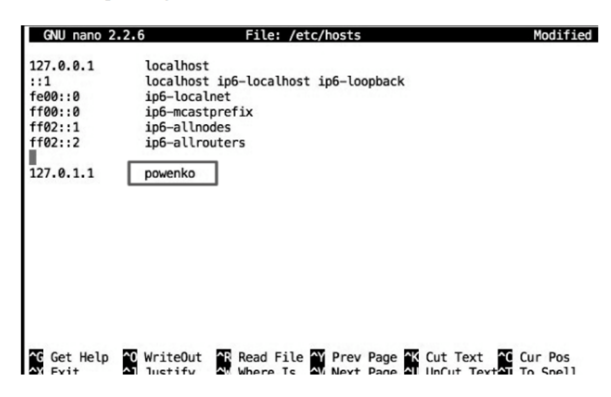


图 23-57　改成其他的名称

STEP2：修改 /etc/hostname

再通过文本编辑器，修改 sudo nano /etc/hostname 的配置文件。

sudo nano /etc/hostname

把 127.0.1.1 后的 raspberry 改成 STEP1 中的名称，如图 23-58 所示。



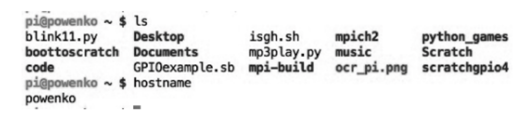


图 23-58 改成 STEP1 中的名称

完成后，按下 Ctrl+O 组合键存储，Ctrl+X 组合键离开文本编辑器 nano。

STEP3：执行 shell 和重新开机

输入以下命令，并重新开机。

```
$ sudo /etc/init.d/hostname.sh
$ sudo reboot
```

3. 结果

重新开机后，hostname 就变换成新的名字了，如图 23-59 所示。

```
pi@powenko ~ $ ls
blink11.py      Desktop        isgh.sh     mpich2       python_games
boottoscratch   Documents      mp3play.py  music        Scratch
code            GPIOexample.sb mpi-build   ocr_pi.png   scratchgpio4
pi@powenko ~ $ hostname
powenko
```

图 23-59 新的 hostname

4. 教学影片

完整的教学视频可以看 RaspberryPi_ChangeHostName。

23.5.3 创建 SSH 钥匙

案例 124

1. 步骤

STEP1：主机创建钥匙

在主机的那台树莓派上输入以下命令，创建 SSH 钥匙，并且在密码 passphrase 的部分，直接按下 Enter 键不要输入任何信息，确认密码 passphrase 的部分，也是一样不要输入任何信息，如图 23-60 所示。

```
$ ssh-keygen -t rsa -C "raspberrypi@raspberrypi"
```

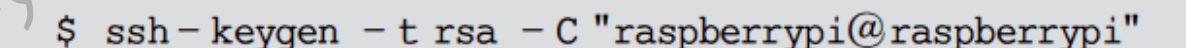
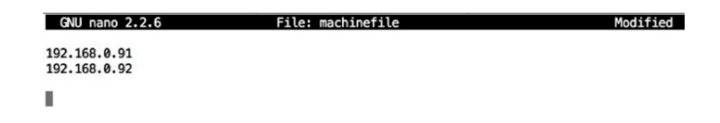


图 23-60　创建 SSH 钥匙

STEP2：把主机钥匙分发给其他的树莓派




```
^G Get Help   ^O WriteOut   ^R Read File   ^Y Prev Page   ^K Cut Text     ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is    ^V Next Page   ^U UnCut Text   ^T To Spell
```

图 23-63　机器本身和其他树莓派的 IP 地址

请再执行一次 mpiexec,可以输入以下命令测试 mpiexec,如图 23-64 所示。

```
$ cd /home/rpimpi/mpi - install/bin
$ mpiexec - f machinefile - n 2 ~/mpi - build/examples/cpi
```

```
pi@raspberrypi /home/rpimpi/mpi-install/bin $ mpiexec -f machinefile -n 2 ~/mpi-build/exa
mples/cpi
Process 1 of 2 is on raspberrypi
Process 0 of 2 is on raspberrypi
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
wall clock time = 0.018717
pi@raspberrypi /home/rpimpi/mpi-install/bin $ sudo nano machinefile
pi@raspberrypi /home/rpimpi/mpi-install/bin $ mpiexec -f machinefile -n 2 ~/mpi-build/exa
mples/cpi
Process 0 of 2 is on raspberrypi
Process 1 of 2 is on powenko
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
wall clock time = 0.010285
```

图 23-64　执行 C 语言的结果

看到的执行时间,会由 0.0187 加快到 0.0102。

2. 教学影片

完整的教学视频可以看 RaspberryPi_mpich3。

注意

目前 mpiexec 只可以执行 C 语言和 Fortran 语言,但是树莓派的程序很多是用 Python,那就需要使用 MPI4PY 软件,如图 23-65 所示,来增加 Python 程序执行的效率。

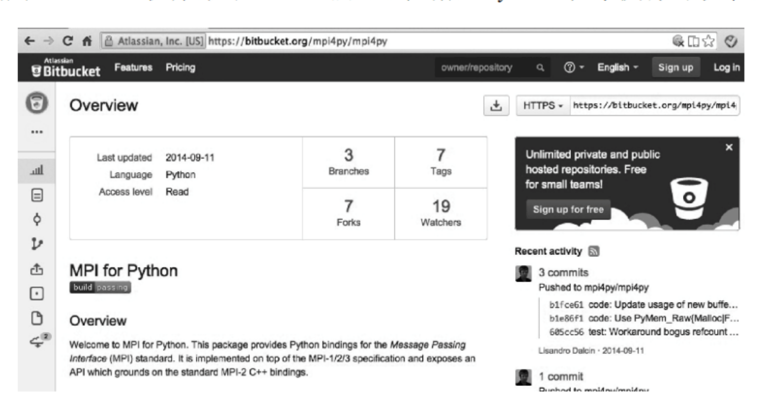


图 23-65　MPI4PY 软件的官方网站